Дж. Прескилл

Квантовая информация и квантовые вычисления

Том 2

R&C
Dynamics


Институт
Компьютерных
Исследований


R&C
Dynamics

Lecture Notes for Physics 229: Quantum Information and Computation

John Preskill
California Institute of Technology

September, 1998

Дж. Прескилл

Квантовая информация и квантовые вычисления

Том 2

Перевод с английского
Т. С. Нечаевой

Под научной редакцией
С. Г. Новокшонова

Москва ♦ Ижевск

2011

УДК 22.314.1
ББК 517.958:530.145.6
П 73

Прескилл Дж.
Квантовая информация и квантовые вычисления. Том 2. — М.–Ижевск: НИЦ «Регулярная и хаотическая динамика», Ижевский институт компьютерных исследований, 2011. — 312 с.

Книга Дж. Прескилла представляет собой наиболее полное современное введение в новую, бурно развивающуюся область науки — теорию квантовой информации и квантовых вычислений.

Вопросы, рассматриваемые во втором томе, объединяет общая тема: защита квантовой информации от ошибок, возникающих как во время ее хранения и передачи, так и при оперировании с ней. В первой из двух основных глав излагаются принципы детектирования, диагностики и коррекции квантовых ошибок; основные типы и принципы организации и работы квантовых кодов коррекции ошибок. Кроме этого в Приложении помещены две обзорные статьи Дж. Прескилла, в которых обсуждается проблема реализации отказоустойчивых квантовых вычислений на основе схем, использующих «шумящие» вентили.

В отдельной большой главе впервые в русскоязычной литературе рассматривается принципиально новый *физический* подход к проблеме защиты квантовой информации от ошибок, в основе которого лежит топологическая устойчивость некоторых квантовых состояний, реализующихся в низкоразмерных многочастичных сильнокоррелированных системах. Несмотря на сложность обсуждаемых в этой главе физических и математических идей, ее содержание дает «ясное представление о предмете без доходящих до абсурда упрощений» и вполне доступно читателю, знакомому с нерелятивистской квантовой механикой, основами теории представлений групп и самыми элементарными сведениями из топологии.

ISBN 978-5-4344-0030-5 **ББК 517.958:530.145.6**

http://shop.rcd.ru
http://ics.org.ru

Оглавление

Глава 7

Коррекция квантовых ошибок

7.1. Квантовые коды коррекции ошибок

Изучая квантовые алгоритмы, мы нашли убедительные свидетельства того, что квантовый компьютер может обладать исключительными способностями. Но будет ли он действительно работать? Сможем ли мы когда-нибудь создать квантовый компьютер и управлять им?

Чтобы добиться этого, необходимо решить проблему защиты квантовой информации от ошибок. Как уже отмечалось в первой главе, у этой проблемы есть несколько аспектов. Между квантовым компьютером и его окружением неизбежно взаимодействие, приводящее к декогерентизации и, следовательно, к разрушению хранящейся в нем квантовой информации. Пока мы не сможем успешно противостоять декогерентизации, наш квантовый компьютер безусловно обречен. Даже если бы нам удалось предотвратить декогерентизацию, полностью изолировав компьютер от окружающей среды, ошибки по-прежнему представляли бы серьезные трудности. Квантовые вентили (в отличие от классических) представляют собой унитарные преобразования, множество которых образует континуум. Следовательно, идеально точное выполнение квантовых операций невозможно. Малые погрешности вентилей будут аккумулироваться, приводя в конце концов к серьезному сбою в вычислении. Любая эффективная стратегия борьбы с ошибками в квантовом компьютере должна обеспечивать защиту как от декогерентизации, так и от малых унитарных ошибок в квантовых схемах.

В этой и следующей главах мы увидим, как искусное кодирование квантовой информации может (в принципе) защитить ее от ошибок. В этой главе будет представлена теория квантовых кодов коррекции ошибок. Мы узнаем, что соответствующим образом закодированная квантовая информация может быть помещена в квантовое запоминающее устройство (квантовую память), подвергаемое разрушительному воздействию шума окружающей среды, и извлечена оттуда неповрежденной (если шум не слишком

силен). Затем в восьмой главе мы распространим эту теорию в двух важных направлениях. Мы увидим, что процедура восстановления информации, может эффективно работать, даже если в ходе ее время от времени случаются ошибки. Мы узнаем, как следует обращаться с закодированной информацией, чтобы квантовые вычисления могли успешно выполняться, несмотря на разрушительное действие декогерентизации и несовершенство квантовых логических вентилей.[1]

Квантовый код коррекции ошибок (КККО) можно рассматривать как отображение k кубитов (гильбертово пространство размерности 2^k) на n кубитов (гильбертово пространство размерности 2^n), где $n > k$. Эти k кубитов представляют собой «логические кубиты» или «закодированные кубиты», которые мы хотим защитить от ошибок. Дополнительные $n-k$ кубитов позволяют хранить k логических кубитов в избыточном виде, так чтобы закодированную информацию было нелегко разрушить.

Для того чтобы лучше понять идею КККО, вернемся к рассмотренному в первой главе примеру кода Шора с $n = 9$ и $k = 1$. Его можно описать, определив два базисных состояния подпространства кода; будем обозначать эти базисные состояния как $|\bar{0}\rangle$ — «логический нуль» и $|\bar{1}\rangle$ — «логическая единица». Они имеют вид

$$\begin{aligned} |\bar{0}\rangle &= \left[\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)\right]^{\otimes 3}, \\ |\bar{1}\rangle &= \left[\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)\right]^{\otimes 3}; \end{aligned} \tag{7.1}$$

каждое базисное состояние представляет собой троекратно повторенное трехкубитовое «кот-состояние». Как вы помните из обсуждения «кот-состояния» в четвертой главе, два базисных состояния можно различить с помощью трехкубитовой наблюдаемой $\boldsymbol{\sigma}_x^1 \otimes \boldsymbol{\sigma}_x^2 \otimes \boldsymbol{\sigma}_x^3$ (где $\boldsymbol{\sigma}_x^i$ обозначает матрицу Паули $\boldsymbol{\sigma}_x$, действующую на i-й кубит); мы будем использовать обозначение $\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3$ для этого оператора. (Здесь подразумевается, что на остальные кубиты действует скрытый в этом обозначении оператор $1 \otimes 1 \otimes \ldots \otimes 1$.) Состояния $|\bar{0}\rangle$ и $|\bar{1}\rangle$ являются собственными векторами оператора $\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3$ с собственными значениями $+1$ и -1 соответственно. Однако невозможно

[1] Материал главы, посвященной отказоустойчивым квантовым вычислениям, до настоящего времени не опубликован. Чтобы как-то компенсировать этот пробел и познакомить читателей с основными принципами реализации отказоустойчивых квантовых вычислений, с разрешения автора в приложении предлагаются переводы двух его обзорных статей, посвященных этой теме. — *Прим. ред.*

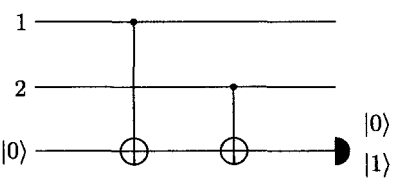
отличить $|\bar{0}\rangle$ от $|\bar{1}\rangle$ (извлечь любую информацию о значении логического кубита), наблюдая любые один или два кубита из имеющихся девяти. В этом смысле, логический кубит закодирован *нелокальным образом*; фактически, он записан в запутанности между кубитами блока. Это свойство нелокальности закодированной информации обеспечивает защиту от шума, если, конечно, он является локальным (то есть действует независимо, или почти независимо, на различные кубиты в блоке).

Предположим, что приготовлено неизвестное квантовое состояние и закодировано как $a|\bar{0}\rangle + b|\bar{1}\rangle$. Пусть теперь возникла ошибка; мы должны выявить ее и уничтожить. Как нам поступить? Допустим для начала, что происходит однократное инвертирование, действующее на один из трех первых кубитов. Тогда, как обсуждалось в первой главе, положение инвертированного кубита можно определить путем измерения двухкубитовых операторов

$$\mathbf{Z}_1\mathbf{Z}_2, \qquad \mathbf{Z}_2\mathbf{Z}_3. \tag{7.2}$$

Базисные логические состояния $|\bar{0}\rangle$ и $|\bar{1}\rangle$ являются собственными векторами этих операторов с собственным значением $+1$. Но инвертирование любого из трех кубитов меняет эти собственные значения. Например, если $\mathbf{Z}_1\mathbf{Z}_2 = -1$, а $\mathbf{Z}_2\mathbf{Z}_3 = 1$,[1] то мы делаем вывод, что инвертирован первый кубит относительно двух других. Мы можем исправить ошибку, еще раз инвертируя этот кубит.

Важно, что наше измерение, диагностирующее инвертированный кубит, является коллективным измерением двух кубитов одновременно — мы узнаем значение $\mathbf{Z}_1\mathbf{Z}_2$, но не должны определять индивидуальные значения $\mathbf{Z}_1$ и $\mathbf{Z}_2$, поскольку это может повредить закодированное состояние. Как выполнить такое коллективное измерение? Фактически, его можно осуществить, располагая квантовым компьютером, способным выполнять операции CNOT (контролируемое НЕ). Сначала мы вводим приготовленный в состоянии $|0\rangle$ дополнительный «служебный» кубит, затем выполняем квантовую схему



[1] Вновь (см. подстрочное примечание на с. 201 первого тома) подобные равенства следует понимать как символические. В данном случае, например, первое из них означает, что состояние с одним инвертированным кубитом является собственным состоянием оператора $\mathbf{Z}_1\mathbf{Z}_2$ с собственным значением -1. — *Прим. ред.*

и, наконец, измеряем служебный кубит. Если кубиты 1 и 2 находятся в состоянии с $\mathbf{Z}_1\mathbf{Z}_2 = -1$ ($|0\rangle_1|1\rangle_2$ или $|1\rangle_1|0\rangle_2$), то служебный кубит однократно инвертируется и результатом его измерения будет $|1\rangle$. Но если кубиты 1 и 2 находятся в состоянии с $\mathbf{Z}_1\mathbf{Z}_2 = +1$ ($|0\rangle_1|0\rangle_2$ или $|1\rangle_1|1\rangle_2$), то служебный кубит останется неизменным или инвертируется дважды, а результатом измерения будет $|0\rangle$. Аналогично, можно выполнить измерение и операторов других двухкубитовых наблюдаемых

$$\begin{array}{ll} \mathbf{Z}_4\mathbf{Z}_5, & \mathbf{Z}_5\mathbf{Z}_6, \\ \mathbf{Z}_7\mathbf{Z}_8, & \mathbf{Z}_8\mathbf{Z}_9, \end{array} \tag{7.3}$$

чтобы диагностировать ошибки инвертирования кубита в двух других кластерах из трех кубитов.

Трехкубитового кода достаточно для защиты от однократного инвертирования бита. Для защиты от фазовых ошибок требуется троекратное повторение трехкубитовых кластеров. Предположим, что возникает фазовая ошибка

$$|\psi\rangle \rightarrow \mathbf{Z}|\psi\rangle, \tag{7.4}$$

действующая на один из девяти кубитов. Мы можем определить, в каком из кластеров она возникла, измерив две шестикубитовые наблюдаемые

$$\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4\mathbf{X}_5\mathbf{X}_6, \qquad \mathbf{X}_4\mathbf{X}_5\mathbf{X}_6\mathbf{X}_7\mathbf{X}_8\mathbf{X}_9. \tag{7.5}$$

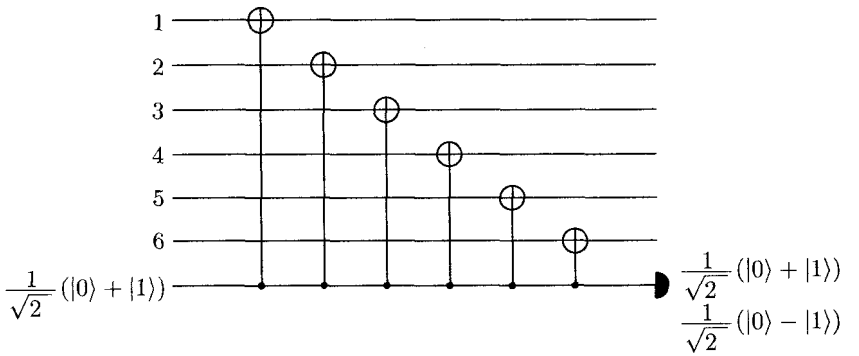
Оба базисных логических состояния $|\bar{0}\rangle$ и $|\bar{1}\rangle$ являются собственными векторами этих операторов с собственным значением одной из этих наблюдаемых (± 1). Фазовая ошибка, действующая на любой один из кубитов в некотором кластере, изменит в нем значение $\mathbf{XXX}$ относительно двух других кластеров; положение этого изменения можно определить путем измерения наблюдаемых (7.5). Как только поврежденный кластер определен, ошибку можно исправить, применяя $\mathbf{Z}$ к одному из кубитов этого кластера.

Как измерить шестикубитовую наблюдаемую $\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4\mathbf{X}_5\mathbf{X}_6$? Заметим, что если начальным состоянием управляющего кубита является $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, а управляемым кубитом является собственное состояние $\mathbf{X}$ (то есть NOT), то вентиль CNOT действует в соответствии с правилом

$$\mathrm{CNOT} : \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |x\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle) \otimes |x\rangle; \tag{7.6}$$

он действует тривиально, если управляемое состояние соответствует собственному значению $\mathbf{X} = +1$ ($x = 0$), и обращает фазу управляющего

кубита, если управляемому соответствует собственное значение $\mathbf{X} = -1$ ($x = 1$).[1] Чтобы измерить произведение $\mathbf{X}$-операторов, мы выполняем схему



а затем измеряем служебный кубит в базисе $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$.

Таким образом, действующая на любой один из девяти кубитов в блоке ошибка не вызывает непоправимого повреждения. Но если в одном кластере из трех кубитов инвертируются два, то закодированная информация *будет* разрушена. Например, если в кластере одновременно инвертируются два первых кубита, то мы неверно определим ошибку и, пытаясь исправить ее, инвертируем третий кубит. Эти ошибки совместно с неправильной попыткой исправления действуют на кодовый блок оператором $\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3$. Поскольку $|\bar{0}\rangle$ и $|\bar{1}\rangle$ являются собственными состояниями оператора $\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3$ с различными собственными значениями, то в результате инвертирования двух битов в одном кластере в закодированном кубите возникает *фазовая ошибка*

$$\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3 \,:\, a|\bar{0}\rangle + b|\bar{1}\rangle \;\rightarrow\; a|\bar{0}\rangle - b|\bar{1}\rangle. \tag{7.7}$$

Закодированная информация также будет повреждена, если фазовые ошибки возникнут в двух разных кластерах. Тогда в результате неверной попытки исправления мы внесем фазовую ошибку в третий кластер, так что в итоге будет применен оператор $\mathbf{Z}_1\mathbf{Z}_4\mathbf{Z}_7$, который инвертирует закодированный кубит

$$\mathbf{Z}_1\mathbf{Z}_4\mathbf{Z}_7 \,:\, a|\bar{0}\rangle + b|\bar{1}\rangle \;\rightarrow\; a|\bar{1}\rangle + b|\bar{0}\rangle. \tag{7.8}$$

[1] Другими словами, в базисе собственных состояний оператора $\mathbf{X}$ управляющий (контролирующий, или источник) и управляемый (контролируемый или целевой) кубиты вентиля CNOT меняются ролями. В том, что это действительно так, нетрудно убедиться, используя его определение (4.11) в базисе собственных состояний оператора Z. Напомним, что в последнем случае под действием CNOT, в зависимости от состояния управляющего кубита изменяется (инвертируется) или остается неизменным управляемый кубит. – *Прим. ред.*

Если вероятность ошибки достаточно мала и если ошибки, действующие на разные кубиты, не сильно скоррелированы, то использование девятикубитового кода позволяет сохранить неизвестный кубит надежнее, чем в том случае, когда мы вообще не беспокоимся о его кодировании. Предположим, например, что на каждый из девяти кубитов окружающая среда действует как описанный в третьей главе деполяризующий канал с вероятностью ошибки p. Тогда вероятность инвертирования бита равна $\frac{2}{3}p$, а вероятность обращения фазы — $\frac{2}{3}p$. (Вероятность одновременного появления обеих ошибок равна $\frac{1}{3}p$.) Нетрудно видеть, что вероятность фазовой ошибки, непоправимо искажающей логический кубит, ограничена сверху величиной $4p^2$, а вероятность подобной ошибки инвертирования бита — величиной $12p^2$. Полная вероятность ошибки не превышает $16p^2$; то есть улучшение по сравнению с вероятностью ошибки p незащищенного кубита имеет место при условии $p < 1/16$.

Конечно, в приведенном выше анализе по умолчанию предполагалось, что кодирование, декодирование, измерение синдрома ошибки и ее исправление выполняются идеально точно. Более реалистический случай, когда ошибки возникают и во время этих операций, обсуждается в приложении.

7.2. Критерии исправления квантовых ошибок

При обсуждении исправления ошибок с помощью девятикубитового кода предполагалось, что каждый кубит подвержен либо ошибке инвертирования бита, либо ошибке обращения фазы (или им обеим). Это нереалистическая модель ошибок, и нам следует понять, как осуществлять коррекцию квантовых ошибок в более общих условиях.

Рассмотрим сначала один кубит, первоначально находящийся в чистом состоянии и произвольным образом взаимодействующий со своим окружением. Из третьей главы мы знаем, что без потери общности (мы все еще можем представлять, что на наш кубит действует самый общий супероператор) можно предполагать, что начальным состоянием окружения является чистое состояние, которое мы обозначим как $|0\rangle_E$. Тогда эволюция кубита и его окружения может быть описана унитарным преобразованием

$$\mathbf{U}: \begin{array}{l} |0\rangle \otimes |0\rangle_E \rightarrow |0\rangle \otimes |e_{00}\rangle_E + |1\rangle \otimes |e_{01}\rangle_E, \\ |1\rangle \otimes |0\rangle_E \rightarrow |0\rangle \otimes |e_{10}\rangle_E + |1\rangle \otimes |e_{11}\rangle_E; \end{array} \tag{7.9}$$

здесь $|e_{ij}\rangle_E$ — четыре состояния окружения, которым не обязательно быть нормированными или взаимно ортогональными (хотя они удовлетворяют

некоторым ограничениям, вытекающим из унитарности $\mathbf{U}$). Под действием $\mathbf{U}$ произвольное состояние кубита $|\psi\rangle = a|0\rangle + b|1\rangle$ эволюционирует как

$$\begin{aligned}\mathbf{U}\colon (a|0\rangle + b|1\rangle) \otimes |0\rangle_E \to\ & a(|0\rangle \otimes |e_{00}\rangle_E + |1\rangle \otimes |e_{01}\rangle_E) + \\ & + b(|0\rangle \otimes |e_{10}\rangle_E + |1\rangle \otimes |e_{11}\rangle_E) = \\ & = (a|0\rangle + b|1\rangle) \otimes \frac{1}{2}(|e_{00}\rangle_E + |e_{11}\rangle_E) + \\ & + (a|1\rangle + b|0\rangle) \otimes \frac{1}{2}(|e_{01}\rangle_E + |e_{10}\rangle_E) + \\ & + (a|1\rangle - b|0\rangle) \otimes \frac{1}{2}(|e_{01}\rangle_E - |e_{10}\rangle_E) + \\ & + (a|0\rangle - b|1\rangle) \otimes \frac{1}{2}(|e_{00}\rangle_E - |e_{11}\rangle_E) \equiv \\ & \equiv \mathbf{1}|\psi\rangle \otimes |e_I\rangle_E + \mathbf{X}|\psi\rangle \otimes |e_X\rangle_E + \\ & + \mathbf{Y}|\psi\rangle \otimes |e_Y\rangle_E + \mathbf{Z}|\psi\rangle \otimes |e_Z\rangle_E. \end{aligned} \tag{7.10}$$

Действие $\mathbf{U}$ может быть разложено по (унитарным) операторам Паули $\{\mathbf{1}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$, просто потому, что они образуют базис в векторном пространстве 2×2-матриц. Эвристически мы можем интерпретировать это разложение, говоря, что с кубитом происходит одно из четырех возможных событий: ничего ($\mathbf{1}$), инвертирование бита ($\mathbf{X}$), обращение фазы ($\mathbf{Z}$) или обе ошибки ($\mathbf{Y} = i\mathbf{XZ}$). Однако не следует понимать эту классификацию буквально, поскольку, пока состояния окружения $\{|e_I\rangle_E, |e_X\rangle_E, |e_Y\rangle_E, |e_Z\rangle_E\}$ не являются взаимно ортогональными, не существует мыслимого измерения, которое могло бы идеально различить эти четыре альтернативы.

Аналогично, действующая в n-кубитовом гильбертовом пространстве произвольная $2^n \times 2^n$-матрица может быть разложена по 2^{2n} операторам

$$\{\mathbf{1}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}^{\otimes n}; \tag{7.11}$$

то есть каждый такой оператор может быть представлен как тензорное произведение однокубитовых операторов, каждый из которых выбирается из единичного $\mathbf{1}$ и трех матриц Паули $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$. Таким образом, действие произвольного унитарного оператора на n кубитов и их окружение можно представить в виде разложения

$$|\psi\rangle \otimes |0\rangle_E \to \sum_a \mathbf{E}_a|\psi\rangle \otimes |e_a\rangle_E; \tag{7.12}$$

здесь индекс a пробегает 2^{2n} значений. $\{\mathbf{E}_a\}$ — множество всех линейно независимых операторов Паули, действующих на n кубитов, а $|e_a\rangle_E$ — со-

ответствующие состояния окружения (которые *не предполагаются* нормированными или взаимно ортогональными). Важной для дальнейшего особенностью этого разложения является то, что каждый оператор $\mathbf{E}_a$ является унитарным.

Уравнение (7.12) обеспечивает идейную основу коррекции квантовых ошибок. При разработке ККК0 мы определяем подмножество $\mathcal{E}$ всех операторов Паули

$$\mathcal{E} \subseteq \{\mathbf{E}_a\} \equiv \{\mathbf{1}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}^{\otimes n}; \tag{7.13}$$

это множество тех ошибок, которые мы хотим уметь исправлять. Нашей целью является выполнение коллективного измерения n кубитов в кодовом блоке, которое позволяет определить, какая из ошибок $\mathbf{E}_a \in \mathcal{E}$ возникла. Если $|\psi\rangle$ — состояние, принадлежащее кодовому подпространству, то для некоторых (но не для всех) кодов это измерение приготовит состояние $\mathbf{E}_a|\psi\rangle \otimes |e_a\rangle_E$, где значение a известно из результата измерения. Так как оператор $\mathbf{E}_a$ является унитарным (и одновременно самосопряженным), мы можем применить к кодовому блоку оператор $\mathbf{E}_a^\dagger$ $(= \mathbf{E}_a)$, восстанавливая неповрежденное состояние $|\psi\rangle$.

Каждому оператору Паули можно сопоставить *вес*, целое число t, удовлетворяющее неравенству $0 \leqslant t \leqslant n$; вес представляет собой количество кубитов, на которые действует нетривиальная матрица Паули ($\mathbf{X}$, $\mathbf{Y}$ или $\mathbf{Z}$). Тогда эвристически слагаемое разложения (7.12), в котором оператор $\mathbf{E}_a$ имеет вес t, можно интерпретировать как событие, состоящее в появлении ошибок в t кубитах (и вновь не следует принимать эту интерпретацию буквально, если состояния $\{|e_a\rangle_E\}$ не являются взаимно ортогональными). Как правило, в качестве $\mathcal{E}$ выбирается совокупность всех операторов Паули с весами вплоть до некоторого t включительно; тогда если удается восстановить исходное состояние после действия на него любого супероператора ошибки с носителем из множества $\mathcal{E}$, то мы говорим, что код может исправить t ошибок. Такой выбор множества $\mathcal{E}$ неявно предполагает, что ошибки, возмущающие разные кубиты, слабо коррелируют между собой, поэтому вероятность возникновения большего, чем t, количества ошибок на n кубитах относительно мала.

Каким необходимым и достаточным условиям должно удовлетворять кодовое подпространство, для того чтобы было возможно исправление заданного множества ошибок $\mathcal{E}$? Обозначим как $\{|\bar{i}\rangle\}$ ортонормированный базис в кодовом подпространстве. (Будем говорить об этих базисных элементах как о «кодовых словах».) Очевидно, *необходимо*, чтобы

$$\langle \bar{j}|\mathbf{E}_b^\dagger \mathbf{E}_a|\bar{i}\rangle = 0, \qquad i \neq j, \tag{7.14}$$

где $\mathbf{E}_{a,b} \in \mathcal{E}$. Если бы это условие не выполнялось для некоторых $i \neq j$, то ошибки могли бы разрушить идеальную различимость ортогональных кодовых слов и закодированной квантовой информации наверняка был бы нанесен ущерб. (Более подробный вывод этого необходимого условия будет представлен ниже.) Также нетрудно видеть, что *достаточным* условием является

$$\langle \bar{j}|\mathbf{E}_b^\dagger \mathbf{E}_a|\bar{i}\rangle = \delta_{ab}\delta_{ij}. \tag{7.15}$$

В этом случае операторы $\mathbf{E}_a$ разбивают кодовое подпространство на совокупность взаимно ортогональных «подпространств ошибок»

$$\mathcal{H}_a = \mathbf{E}_a \mathcal{H}_{code}. \tag{7.16}$$

Предположим, что в произвольное состояние $|\psi\rangle$, приготовленное в кодовом пространстве, вкралась ошибка. Тогда итоговым состоянием кодового блока и окружения является

$$\sum_{\mathbf{E}_a \in \mathcal{E}} \mathbf{E}_a|\psi\rangle \otimes |e_a\rangle_E, \tag{7.17}$$

где суммирование ведется по ошибкам, принадлежащим множеству $\mathcal{E}$. В этом случае можно выполнить ортогональное измерение, проецирующее кодовый блок на одно из пространств $\mathcal{H}_a$, так что состояние приобретает вид

$$\mathbf{E}_a|\psi\rangle \otimes |e_a\rangle_E. \tag{7.18}$$

Наконец, для завершения процедуры восстановления к кодовому блоку применяется унитарный оператор $\mathbf{E}_a^\dagger$.

Код, удовлетворяющий условию (7.15), называется *невырожденным*. Этот термин означает, что существует измерение, которое может однозначно выявить возникшую ошибку $\mathbf{E}_a \in \mathcal{E}$. Но пример девятикубитового кода только что показал, что возможны более общие коды. Девятикубитовый код *вырожден*, поскольку фазовые ошибки, действующие на разные кубиты одного и того же кластера из трех кубитов, одинаковым образом влияют на кодовое подпространство (например, $\mathbf{Z}_1|\psi\rangle = \mathbf{Z}_2|\psi\rangle$). Хотя ни одно измерение не может определить, в каком из кубитов возникла ошибка, это не является помехой для ее успешного исправления.

Нетрудно установить необходимое и достаточное условие возможности восстановления

$$\langle \bar{j}|\mathbf{E}_b^\dagger \mathbf{E}_a|\bar{i}\rangle = C_{ab}\delta_{ij}, \tag{7.19}$$

где $\mathbf{E}_{a,b} \in \mathcal{E}$, а $C_{ab} = \langle \bar{i}|\mathbf{E}_b^\dagger \mathbf{E}_a|\bar{i}\rangle$ — произвольная эрмитова матрица. Нетривиальное содержание этого условия, которое существенно сильнее более

слабого необходимого условия (7.14), состоит в том, что $\langle\bar{i}|\mathbf{E}_b^\dagger\mathbf{E}_a|\bar{i}\rangle$ не зависит от i. Природа этого условия очевидна — будь это иначе, при определении подпространства ошибки $\mathcal{H}_a$ мы получали бы некоторую информацию о закодированном состоянии, что неизбежно приводило бы к его возмущению.

Чтобы доказать необходимость и достаточность условия (7.19), обратимся к развитой в третьей главе теории супероператоров. Действующая на кодовый блок ошибка описывается супероператором, и проблема состоит в том, можно ли построить другой супероператор (процедура восстановления), аннулирующий ее действие. В третьей главе мы узнали, что обратить можно только те супероператоры, которые являются унитарными операторами. Однако от нас не требуется умение аннулировать действие супероператора ошибки на любое состояние в n-кубитовом кодовом блоке; вполне достаточно уметь исправлять ошибки в состояниях, первоначально принадлежавших k-кубитовому закодированному подпространству.

Альтернативным выражением действия ошибки на одно из кодовых базисных состояний $|\bar{i}\rangle$ (и на окружение) является

$$|\bar{i}\rangle\otimes|0\rangle_E \rightarrow \sum_\mu \mathbf{M}_\mu|\bar{i}\rangle\otimes|\mu\rangle_E, \tag{7.20}$$

где теперь состояния $|\mu\rangle_E$ представляют собой элементы *ортонормированного базиса* окружения, а матрицы $\mathbf{M}_\mu$ являются линейными комбинациями содержащихся в $\mathcal{E}$ операторов Паули $\mathbf{E}_a$ и удовлетворяют условию нормировки операторной суммы

$$\sum_\mu \mathbf{M}_\mu^\dagger\mathbf{M}_\mu = \mathbf{1}. \tag{7.21}$$

Ошибка может быть исправлена оператором восстановления, если существуют такие операторы $\mathbf{R}_\nu$, что

$$\sum_\nu \mathbf{R}_\nu^\dagger\mathbf{R}_\nu = \mathbf{1} \tag{7.22}$$

и

$$\sum_{\mu,\nu} \mathbf{R}_\nu\mathbf{M}_\mu|\bar{i}\rangle\otimes|\mu\rangle_E\otimes|\nu\rangle_A = |\bar{i}\rangle\otimes|\mathrm{stuff}\rangle_{E,A}; \tag{7.23}$$

здесь векторы $|\nu\rangle_A$ являются элементами ортонормированного базиса гильбертова пространства служебного кубита, привлекаемого для осуществления операции восстановления, а состояние окружающей среды и служебного кубита $|\mathrm{stuff}\rangle_{E,A}$ не должно зависеть от i. Отсюда следует, что для каждого μ и ν

$$\mathbf{R}_\nu\mathbf{M}_\mu|\bar{i}\rangle = \lambda_{\nu\mu}|\bar{i}\rangle; \tag{7.24}$$

в кодовом подпространстве действие произведения $\mathbf{R}_\nu \mathbf{M}_\mu$ эквивалентно умножению на число. Используя условие нормировки, которому удовлетворяют операторы $\mathbf{R}_\nu$, мы находим, что

$$\mathbf{M}_\delta^\dagger \mathbf{M}_\mu |\bar{i}\rangle = \mathbf{M}_\delta^\dagger \left(\sum_\nu \mathbf{R}_\nu^\dagger \mathbf{R}_\nu \right) \mathbf{M}_\mu |\bar{i}\rangle = \sum_\nu \lambda_{\nu\delta}^* \lambda_{\nu\mu} |\bar{i}\rangle, \tag{7.25}$$

так что действие $\mathbf{M}_\delta^\dagger \mathbf{M}_\mu$ в кодовом подпространстве также эквивалентно умножению на число. Другими словами,

$$\langle \bar{j} | \mathbf{M}_\delta^\dagger \mathbf{M}_\mu | \bar{i} \rangle = C_{\delta\mu} \delta_{ij}; \tag{7.26}$$

отсюда следует (7.19), поскольку каждый оператор $\mathbf{E}_a$ из $\mathcal{E}$ является линейной комбинацией операторов $\mathbf{M}_\mu$.

Другой поучительный способ понять, почему (7.26) является необходимым условием возможности исправления ошибки, — это обратить внимание на то, что если кодовый блок приготовлен в состоянии $|\psi\rangle$, а ошибка действует в соответствии (7.20), то получаемая путем вычисления следа по кодовому блоку матрица плотности окружения имеет вид

$$\boldsymbol{\rho}_E = \sum_{\mu\nu} |\mu\rangle_E \langle\psi| \mathbf{M}_\nu^\dagger \mathbf{M}_\mu |\psi\rangle \, {}_E\langle\nu|. \tag{7.27}$$

Ошибка может быть успешно исправлена только в том случае, если в процессе измерения окружения невозможно получить какую-либо информацию о состоянии $|\psi\rangle$. Следовательно, мы требуем, чтобы $\boldsymbol{\rho}_E$ не зависела от $|\psi\rangle$, если $|\psi\rangle$ — произвольное состояние из кодового подпространства; тогда отсюда следует уравнение (7.26).

Чтобы увидеть, что уравнение (7.26) как необходимо, так и достаточно, можно явно построить исправляющий ошибки супероператор. С этой целью достаточно выбрать базис окружения $\{|\mu\rangle_E\}$ таким образом, чтобы матрица $C_{\delta\mu}$ в уравнении (7.26) была диагональна

$$\langle \bar{j} | \mathbf{M}_\delta^\dagger \mathbf{M}_\mu | \bar{i} \rangle = C_\mu \delta_{\delta\mu} \delta_{ij}, \tag{7.28}$$

где $\sum_\mu C_\mu = 1$ вытекает из условия нормировки операторной суммы. Пусть для каждого ν с $C_\nu \neq 0$

$$\mathbf{R}_\nu = \frac{1}{\sqrt{C_\nu}} \sum_i |\bar{i}\rangle \langle \bar{i} | \mathbf{M}_\nu^\dagger, \tag{7.29}$$

так что $\mathbf{R}_\nu$ действует в соответствии с

$$\mathbf{R}_\nu\colon \mathbf{M}_\mu|\bar{i}\rangle \to \sqrt{C_\nu}\delta_{\mu\nu}|\bar{i}\rangle. \tag{7.30}$$

Тогда нетрудно понять, что

$$\sum_{\mu,\nu}\mathbf{R}_\nu\mathbf{M}_\mu|\bar{i}\rangle\otimes|\mu\rangle_E\otimes|\nu\rangle_A = |\bar{i}\rangle\otimes\sum_\nu\sqrt{C_\nu}|\nu\rangle_E\otimes|\nu\rangle_A; \tag{7.31}$$

определяемый $\mathbf{R}_\nu$ супероператор действительно исправляет ошибку. Остается лишь проверить, что $\mathbf{R}_\nu$ удовлетворяют условию нормировки. Имеем

$$\sum_\nu\mathbf{R}_\nu^\dagger\mathbf{R}_\nu = \sum_{\nu,i}\frac{1}{C_\nu}\mathbf{M}_\nu|\bar{i}\rangle\langle\bar{i}|\mathbf{M}_\nu^\dagger, \tag{7.32}$$

что представляет собой ортогональный проектор на пространство состояний, которые достигаются в результате действия ошибок на кодовые слова. Таким образом, мы можем завершить подробное описание супероператора восстановления, добавив к операторной сумме еще один элемент — проектор на дополнительное подпространство.

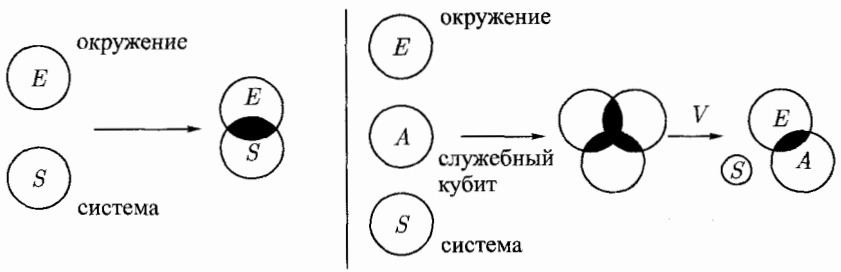
Итак, уравнение (7.19) является достаточным условием исправления ошибок, поскольку для операторов ошибок можно выбрать базис, диагонализирующий матрицу C_{ab} (не обязательно базис операторов Паули), а в этом базисе можно однозначно диагностировать ошибку, выполняя соответствующее ортогональное измерение. (Собственные моды C_{ab} с равными нулю собственными значениями, подобно $\mathbf{Z}_1-\mathbf{Z}_2$ в случае 9-кубитового кода, соответствуют ошибкам, вероятность появления которых равна нулю.) Таким образом, как только совокупность возможных ошибок $\mathcal{E}$ установлена, операция восстановления определена. В частности, не нужна никакая информация о связанных с ошибками $\mathbf{E}_a$ состояниях окружения $|e_a\rangle_E$. Следовательно, код одинаково эффективно борется как с унитарными ошибками, так и с *ошибками декогерентизации* (до тех пор, пока пренебрежимо мала вероятность появления ошибок, не принадлежащих множеству $\mathcal{E}$). Конечно, в случае невырожденного кода C_{ab} диагональна уже в базисе Паули, и мы можем представить базис восстановления в виде

$$\mathbf{R}_a = \sum_i|\bar{i}\rangle\langle\bar{i}|\mathbf{E}_a^\dagger; \tag{7.33}$$

каждому $\mathbf{E}_a$ из $\mathcal{E}$ соответствует $\mathbf{R}_a$.

Мы описали коррекцию ошибок как процедуру, состоящую из двух этапов: во-первых, для выявления ошибки проводится коллективное измерение, а во-вторых, для ее исправления осуществляется обусловленное результатом измерения унитарное преобразование. Эта точка зрения имеет много достоинств. В частности, именно процедура квантового измерения, по-видимому, позволяет укротить континуум возможных ошибок, поскольку измерение проецирует поврежденное состояние на один из дискретного множества результатов, для каждого из которых существует инструкция по восстановлению. Но в действительности измерение — не самый важный этап процесса коррекции квантовых ошибок. Конечно, супероператор восстановления (7.31) может рассматриваться как ортогональное преобразование, действующее на кодовый блок и служебный кубит. Этот супероператор может описывать следующее за унитарным оператором измерение, если мы представим, что служебный кубит подвергается ортогональному измерению, но измерение не является необходимым.

В отсутствие измерения мы можем взглянуть с другой стороны на достигаемое в процессе восстановления обращение декогерентизации. Когда кодовый блок взаимодействует с окружением, он запутывается с ним. В результате неймановская энтропия окружения (как и энтропия кодового блока) возрастает. Если мы не в состоянии управлять окружением, то рост его энтропии никогда не будет обращен; почему в таком случае возможна коррекция квантовых ошибок? Предоставляемый уравнением (7.31) ответ состоит в том, что мы можем применить унитарное преобразование к информации и служебному кубиту, которыми мы *действительно* управляем. Если критерии коррекции квантовых ошибок удовлетворены, то можно выбрать унитарное преобразование, позволяющее запутывание информации с окружением трансформировать в запутывание служебного кубита с окружением, восстанавливая тем самым чистоту информации, как это показано на рисунке.



В то время как измерение не является обязательной составной частью процедуры коррекции ошибок, служебный кубит абсолютно необходим. Он играет роль депозитария для энтропии, вносимой в кодовый блок ошибками — он «разогревается», тогда как информация «охлаждается». Если мы должны в течение длительного времени продолжать защиту квантовой информации, хранящейся в квантовой памяти, то для этой цели необходимо наладить непрерывную поставку служебных кубитов, которые можно отбрасывать после использования. Если же служебный кубит используется повторно, то для этого он должен быть предварительно очищен. Как обсуждалось в первой главе, удаление является диссипативным процессом. Следовательно, согласно принципам термодинамики, коррекция (квантовых) ошибок требует энергетических затрат. Ошибки являются причиной проникновения энтропии в информацию. С помощью обратимого процесса эту энтропию можно перенести на служебный кубит, но для того чтобы откачать ее из служебного кубита и вернуть в окружающую среду, необходимо совершить определенную работу.

7.3. Некоторые основные свойства ККК0

7.3.1. Расстояние

Говорят, что квантовый код является *двоичным*, если он может быть представлен на языке кубитов. В двоичном коде кодовое подпространство размерности 2^k погружено в пространство размерности 2^n, где k и $n\,(>k)$ — целые числа. В сущности, нет никакой необходимости требовать, чтобы размерности этих пространств были степенями двойки (смотри упражнения); тем не менее, мы здесь главным образом будем ограничиваться двоичным кодированием как наиболее простым.

В дополнение к размеру блока n и количеству закодированных кубитов k, еще одним важным параметром, характеризующим код, является расстояние d. Расстояние d представляет собой минимальный вес оператора Паули $\mathbf{E}$, такого, что

$$\langle \bar{i}|\mathbf{E}_a|\bar{j}\rangle \neq C_a\delta_{ij}. \tag{7.34}$$

Квантовый код с размером n, количеством закодированных кубитов k и расстоянием d будет кратко обозначаться символом $[[n,k,d]]$. Обозначение с двойными скобками используется для квантового кода, чтобы отличить его от обозначения $[n,k,d]$ для классического кода.

Будем говорить, что ККК0 может исправить t ошибок, если множество $\mathcal{E}$ допускающих исправление ошибок $\mathbf{E}_a$ включает все операторы Паули с весом, не превышающим t. Наше определение расстояния подразумевает,

что критерию коррекции ошибок

$$\langle \bar{i}|\mathbf{E}_a^\dagger \mathbf{E}_b|\bar{j}\rangle = C_{ab}\delta_{ij} \tag{7.35}$$

удовлетворяют все операторы Паули $\mathbf{E}_a$ с весом, не превышающим t, при условии, что $d \geqslant 2t+1$. Следовательно, ККК0 с расстоянием $d = 2t+1$ может исправить t ошибок.

7.3.2. Локализованные ошибки

Код с расстоянием $d \geqslant 2t+1$ может исправить t ошибок, независимо от их положения в кодовом блоке. Но иногда мы можем знать, что некоторые кубиты особенно предрасположены к появлению ошибок. Возможно, мы видели, как по ним ударили молотком. Или, может быть, вы послали мне блок из n кубитов, но $t\,(<n)$ из них оказались потерянными и уже никогда не будут получены. Я уверен, что остальные $n-t$ кубитов были хорошо упакованы и получены неповрежденными. Тогда я заменяю t недостающих кубитов (произвольно выбранным) состоянием $|00\ldots0\rangle$, вполне отдавая себе отчет в том, что эти кубиты могут содержать ошибки.

Тот же самый код может защитить от большего количества ошибок, если они появляются в известных местах. Фактически ККК0 с расстоянием $d = t+1$ может исправить t ошибок в известных положениях. В этом случае множество ошибок $\mathcal{E}$, которые необходимо исправить, представляет собой совокупность всех операторов Паули с *носителем* в t определенных местоположениях (каждый $\mathbf{E}_a$ действует тривиально на другие $n-t$ кубитов). Но тогда для каждого $\mathbf{E}_a$ и $\mathbf{E}_b$ из $\mathcal{E}$ произведение $\mathbf{E}_a^\dagger \mathbf{E}_b$ также имеет вес не больше, чем t. Следовательно, критерий коррекции ошибок удовлетворяется для всех $\mathbf{E}_{a,b} \in \mathcal{E}$, при условии, что код имеет расстояние, по крайней мере равное $t+1$.

В частности, ККК0, корректирующий t ошибок, находящихся в произвольных положениях, может исправить $2t$ ошибок в известных положениях.

7.3.3. *Обнаружение ошибок*

В некоторых случаях оказывается достаточным просто заметить ошибку, даже если мы не можем полностью диагностировать или исправить ее. Предназначенное для регистрации ошибок измерение имеет два возможных результата: «good» и «bad». Если получается результат «good», мы уверены, что квантовое состояние не повреждено. Если получается результат «bad», значит, состояние было повреждено, и от него следует избавиться.

Если носитель супероператора ошибки принадлежит множеству всех операторов Паули $\mathcal{E}$ с весом, не превышающим t, и возможно измерение, точно показывающее, возникла ошибка или нет, то в таком случае говорят, что мы можем обнаружить t ошибок. Обнаружение ошибок проще, чем их коррекция, поэтому один и тот же код может детектировать больше ошибок, чем исправить. Фактически, ККК0 с расстоянием $d = t + 1$ может обнаружить t ошибок.

Такой код обладает свойством

$$\langle \bar{i}|\mathbf{E}_a|\bar{j}\rangle = C_a\delta_{ij} \tag{7.36}$$

для каждого оператора Паули $\mathbf{E}_a$ с весом не выше t, или

$$\mathbf{E}_a|\bar{i}\rangle = C_a|\bar{i}\rangle + |\varphi_{ai}^{\perp}\rangle, \tag{7.37}$$

где $|\varphi_{ai}^{\perp}\rangle$ — ненормированный вектор, ортогональный кодовому подпространству. Следовательно, действие супероператора ошибки с носителем в $\mathcal{E}$ на состояние кодового подпространства $|\psi\rangle$ имеет вид

$$|\psi\rangle\otimes|0\rangle_E \to \sum_{\mathbf{E}_a\in\mathcal{E}} \mathbf{E}_a|\psi\rangle\otimes|e_a\rangle_E = |\psi\rangle\otimes\sum_{\mathbf{E}_a\in\mathcal{E}} C_a\otimes|e_a\rangle_E + |\mathrm{orthog}\rangle, \tag{7.38}$$

где $|\mathrm{orthog}\rangle$ обозначает вектор, ортогональный кодовому подпространству.

Теперь мы можем выполнить «грубое» ортогональное измерение информации, с двумя результатами: состояние проецируется либо на подпространство кодов, либо на дополнительное ему подпространство. Первый результат воспроизводит неповрежденное состояние $|\psi\rangle$, второй — сообщает об обнаружении ошибки. Таким образом, ККК0 с расстоянием d может регистрировать $d-1$ ошибок. В частности, если ККК0 может исправить t ошибок, то детектировать он может $2t$ ошибок.

7.3.4. Квантовые коды и запутывание

ККК0 защищает квантовую информацию от ошибок, кодируя ее *нелокальным* образом, то есть распределяя ее между несколькими кубитами в блоке. Таким образом, квантовое кодовое слово представляет собой сильно запутанное состояние.

Фактически, невырожденный код с расстоянием $d = t + 1$ обладает следующим свойством. Выберем любое, принадлежащее кодовому подпространству, состояние $|\psi\rangle$ и любые t кубитов в блоке. Возьмем след по оставшимся $n - t$ кубитам, чтобы получить матрицу плотности t кубитов

$$\boldsymbol{\rho}^{(t)} = \mathrm{tr}_{(n-t)}\,|\psi\rangle\langle\psi|. \tag{7.39}$$

Тогда эта матрица плотности абсолютно случайна

$$\boldsymbol{\rho}^{(t)} = \frac{1}{2^t}\mathbf{1}. \tag{7.40}$$

(Наблюдая любые t кубитов в блоке, мы ничего не можем узнать об информации, закодированной кодом с расстоянием $t+1$; то есть $\boldsymbol{\rho}^{(t)}$ — независимая от кодового слова константа. Но матрица плотности t кубитов действительно будет кратной единичному оператору, если только код невырожден.)

Чтобы проверить свойство (7.40), заметим, что для невырожденного кода с расстоянием $t+1$

$$\langle \bar{i}|\mathbf{E}_a|\bar{j}\rangle = 0 \tag{7.41}$$

для любого $\mathbf{E}_a$ ненулевого веса, не превышающего t. Так что

$$\mathrm{tr}(\boldsymbol{\rho}^{(t)}\mathbf{E}_a) = 0, \tag{7.42}$$

для любого, отличного от единичного, t-кубитового оператора Паули. Теперь $\boldsymbol{\rho}^{(t)}$, подобно любой эрмитовой $2^t \times 2^t$-матрице, может быть разложена по операторам Паули:

$$\boldsymbol{\rho}^{(t)} = \frac{1}{2^t}\mathbf{1} + \sum_{\mathbf{E}_a \neq \mathbf{1}} \rho_a \mathbf{E}_a. \tag{7.43}$$

Так как операторы $\mathbf{E}_a$ удовлетворяют условию

$$\frac{1}{2^t}\,\mathrm{tr}(\mathbf{E}_a\mathbf{E}_b) = \delta_{ab}, \tag{7.44}$$

мы находим, что все $\rho_a = 0$, и приходим к выводу, что $\boldsymbol{\rho}^{(t)}$ пропорциональна единичному оператору.

7.4. Вероятность сбоя

7.4.1. Нижняя граница точности воспроизведения

Если носитель супероператора ошибок содержит только операторы Паули из множества $\mathcal{E}$, способ исправления которых нам известен, то закодированная квантовая информация может быть восстановлена с идеальной точностью воспроизведения. Однако в реальной ситуации всегда существует небольшая, но отличная от нуля, вероятность появления ошибок, которые не входят в $\mathcal{E}$, так что восстановленное состояние не будет идеальным. Что можно сказать о точности воспроизведения восстановленного состояния?

Разложение супероператора ошибок по операторам Паули можно разбить на сумму «хороших» (входящих в $\mathcal{E}$) и «плохих» (не входящих в $\mathcal{E}$) операторов. В соответствии с этим результат его действия на состояние кодового подпространства $|\psi\rangle$ можно представить в виде

$$\begin{aligned} |\psi\rangle \otimes |0\rangle_E \to & \sum_a \mathbf{E}_a |\psi\rangle \otimes |e_a\rangle_E = \\ = & \sum_{\mathbf{E}_a \in \mathcal{E}} \mathbf{E}_a |\psi\rangle \otimes |e_a\rangle_E + \sum_{\mathbf{E}_b \notin \mathcal{E}} \mathbf{E}_b |\psi\rangle \otimes |e_b\rangle_E = \\ = & \, |\mathrm{GOOD}\rangle + |\mathrm{BAD}\rangle. \end{aligned} \tag{7.45}$$

Тогда операция восстановления (унитарное преобразование, действующее на информацию и служебный кубит) отображает $|\mathrm{GOOD}\rangle$ на состояние информации, окружающей среды и служебного кубита $|\mathrm{GOOD}'\rangle$, а $|\mathrm{BAD}\rangle$ — на состояние $|\mathrm{BAD}'\rangle$, так что после восстановления мы получаем состояние

$$|\mathrm{GOOD}'\rangle + |\mathrm{BAD}'\rangle; \tag{7.46}$$

здесь (поскольку, действуя на «хорошее» состояние, восстановление работает идеально)

$$|\mathrm{GOOD}'\rangle = |\psi\rangle \otimes |s\rangle_{EA}, \tag{7.47}$$

где $|s\rangle_{EA}$ — некоторое состояние окружения и служебного кубита.

Предположим, что состояния $|\mathrm{GOOD}\rangle$ и $|\mathrm{BAD}\rangle$ взаимно ортогональны. Это справедливо, если, в частности, все «хорошие» состояния окружения ортогональны всем «плохим» состояниям, то есть если

$$\langle e_a | e_b \rangle = 0 \quad \text{при} \quad \mathbf{E}_a \in \mathcal{E},\ \mathbf{E}_b \notin \mathcal{E}. \tag{7.48}$$

Пусть $\boldsymbol{\rho}_{\mathrm{rec}}$ обозначает матрицу плотности восстановленного состояния, полученную путем вычисления следа по состояниям окружения и служебного кубита, и пусть

$$F = \langle \psi | \boldsymbol{\rho}_{\mathrm{rec}} | \psi \rangle \tag{7.49}$$

— его точность воспроизведения. Теперь, поскольку $|\mathrm{BAD}'\rangle$ ортогонально $|\mathrm{GOOD}'\rangle$ (то есть $|\mathrm{BAD}'\rangle$ не имеет ни одной компоненты вдоль $|\psi\rangle \otimes |s\rangle_{EA}$), точность воспроизведения равна

$$F = \langle \psi | \boldsymbol{\rho}_{\mathrm{GOOD}'} | \psi \rangle + \langle \psi | \boldsymbol{\rho}_{\mathrm{BAD}'} | \psi \rangle, \tag{7.50}$$

где

$$\boldsymbol{\rho}_{\mathrm{GOOD}'} = \mathrm{tr}_{EA}(|\mathrm{GOOD}'\rangle\langle\mathrm{GOOD}'|), \qquad \boldsymbol{\rho}_{\mathrm{BAD}'} = \mathrm{tr}_{EA}(|\mathrm{BAD}'\rangle\langle\mathrm{BAD}'|). \tag{7.51}$$

Следовательно, точность воспроизведения восстановленного состояния удовлетворяет неравенству

$$F \geqslant \langle\psi|\rho_{\mathrm{GOOD}'}|\psi\rangle = \||s\rangle_{EA}\|^2 = \||\mathrm{GOOD}'\rangle\|^2. \tag{7.52}$$

Более того, в силу унитарности операции восстановления $\||\mathrm{GOOD}'\rangle\| = \||\mathrm{GOOD}\rangle\|$ и, следовательно,

$$F \geqslant \||\mathrm{GOOD}\rangle\|^2 = \left\| \sum_{\mathbf{E}_a \in \mathcal{E}} \mathbf{E}_a|\psi\rangle \otimes |e_a\rangle_E \right\|^2. \tag{7.53}$$

Однако в общем случае $|\mathrm{BAD}\rangle$ не обязательно ортогонально $|\mathrm{GOOD}\rangle$, так что $|\mathrm{BAD}'\rangle$ не должно быть ортогонально $|\mathrm{GOOD}'\rangle$. Тогда $|\mathrm{BAD}'\rangle$ может иметь компоненту вдоль $|\mathrm{GOOD}'\rangle$, которая деструктивно интерферирует с $|\mathrm{GOOD}'\rangle$ и, следовательно, понижает точность воспроизведения. Тем не менее мы можем получить нижнюю границу точности воспроизведения и в этом, более общем, случае, разлагая $|\mathrm{BAD}'\rangle$ на компоненту вдоль $|\mathrm{GOOD}'\rangle$ и ортогональную компоненту

$$|\mathrm{BAD}'\rangle = |\mathrm{BAD}'_{\|}\rangle + |\mathrm{BAD}'_{\perp}\rangle. \tag{7.54}$$

Тогда, рассуждая так же, как и выше, получаем

$$F \geqslant \||\mathrm{GOOD}'\rangle + |\mathrm{BAD}'_{\|}\rangle\|^2. \tag{7.55}$$

Конечно, так как и операция ошибки, и операция восстановления унитарно действуют на информацию, окружение и служебный кубит, полное состояние $|\mathrm{GOOD}'\rangle + |\mathrm{BAD}'\rangle$ нормировано, или

$$\||\mathrm{GOOD}'\rangle + |\mathrm{BAD}'_{\|}\rangle\|^2 + \||\mathrm{BAD}'_{\perp}\rangle\|^2 = 1, \tag{7.56}$$

а неравенство (7.55) приобретает вид

$$F \geqslant 1 - \||\mathrm{BAD}'_{\perp}\rangle\|^2. \tag{7.57}$$

Наконец, норма вектора $|\mathrm{BAD}'_{\perp}\rangle$ не может превысить норму вектора $|\mathrm{BAD}'\rangle$ и, следовательно,

$$1 - F \leqslant \||\mathrm{BAD}'\rangle\|^2 = \||\mathrm{BAD}\rangle\|^2 \equiv \left\| \sum_{\mathbf{E}_b \notin \mathcal{E}} \mathbf{E}_b|\psi\rangle \otimes |e_b\rangle_E \right\|^2. \tag{7.58}$$

Это наиболее общая нижняя граница «вероятности сбоя» операции восстановления. Уравнение (7.53) следует отсюда в частном случае, когда $|\mathrm{GOOD}\rangle$ и $|\mathrm{BAD}\rangle$ являются взаимно ортогональными состояниями.

7.4.2. Некоррелированные ошибки

Рассмотрим теперь некоторые приложения этих результатов для случая, когда действующие на отдельные кубиты ошибки полностью некоррелированы. В этом случае супероператор ошибок является тензорным произведением однокубитовых супероператоров. Если на самом деле ошибки одинаково действуют на все кубиты, то мы можем представить n-кубитовый супероператор как

$$\$_{\text{error}}^{(n)} = \left[\$_{\text{error}}^{(1)}\right]^{\otimes n}, \tag{7.59}$$

где $\$_{\text{error}}^{(1)}$ — однокубитовый супероператор, действие которого (в его унитарном представлении) имеет вид

$$|\psi\rangle\otimes|0\rangle_E \to |\psi\rangle\otimes|e_I\rangle_E + \mathbf{X}|\psi\rangle\otimes|e_X\rangle_E + \mathbf{Y}|\psi\rangle\otimes|e_Y\rangle_E + \mathbf{Z}|\psi\rangle\otimes|e_Z\rangle_E. \tag{7.60}$$

Влияние ошибок на закодированную информацию особенно легко анализировать, если предположить, что каждое из трех состояний окружения $|e_{X,Y,Z}\rangle$ ортогонально состоянию $|e_I\rangle$. В этом случае запись того, возникла ошибка или нет, для каждого кубита постоянно отпечатывается на окружении, и разумно говорить о вероятности ошибки p_{error} для каждого кубита, где

$$\langle e_I|e_I\rangle = 1 - p_{\text{error}}. \tag{7.61}$$

Если наш квантовый код может исправить t ошибок, то «хорошие» операторы Паули имеют вес, не превышающий t, а «плохие» — вес, превышающий t. Тогда восстановление несомненно будет успешным, по крайней мере пока ошибкам не подвергнутся $t+1$ кубитов. Отсюда следует, что точность воспроизведения подчиняется условию

$$1 - F \leqslant \sum_{s=t+1}^{n} \binom{n}{s} p_{\text{error}}^s (1 - p_{\text{error}})^{n-s} \leqslant \binom{n}{t+1} p_{\text{error}}^{t+1}. \tag{7.62}$$

(Для каждого из $\binom{n}{t+1}$ способов выбора $t+1$ положений, вероятность возникновения ошибки во всех этих позициях равна p_{error}^{t+1}, где мы пренебрегаем вероятностью возникновения дополнительных ошибок в остальных $n-t-1$ позициях. Следовательно, окончательное выражение (7.62) определяет верхний предел вероятности возникновения по крайней мере $t+1$ ошибок в блоке из n кубитов.) При малой p_{error} и большом t точность воспроизведения закодированной информации существенно улучшается по сравнению с точностью воспроизведения $F = 1 - O(p)$ незащищенного кубита.

Для действующего на один кубит общего супероператора ошибок не существует четкого понятия «вероятности ошибки»; состояние кубита и окружения, полученное в результате действия оператора Паули $\mathbf{1}$, не ортогонально (и, следовательно, его нельзя полностью отличить) состоянию, полученному в результате действия операторов Паули $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$. В предельном случае, когда декогерентизация вообще отсутствует, «ошибки» возникают в результате действия на кубиты неизвестных унитарных преобразований. (Если бы действующее на кубит унитарное преобразование $\mathbf{U}$ было известно, то мы могли бы исправить «ошибку», просто применив $\mathbf{U}^\dagger$.)

Рассмотрим некоррелированные унитарные ошибки, действующие на n кубитов в кодовом блоке, каждая из которых (с точностью до несущественной фазы) имеет вид

$$\mathbf{U}^{(1)} = \sqrt{1-p} + i\sqrt{p}\mathbf{W}, \tag{7.63}$$

где $\mathbf{W}$ — (бесследовая, эрмитова) линейная комбинация операторов $\mathbf{X}$, $\mathbf{Y}$ и $\mathbf{Z}$, удовлетворяющая условию $\mathbf{W}^2 = \mathbf{1}$. Если приготовлено состояние кубита $|\psi\rangle$, а затем возникает унитарная ошибка (7.63), то точность воспроизведения итогового состояния

$$F = |\langle\psi|\mathbf{U}^{(1)}|\psi\rangle|^2 = 1 - p(1 - (\langle\psi|\mathbf{W}|\psi\rangle)^2) \geqslant 1 - p. \tag{7.64}$$

Если унитарная ошибка (7.63) действует на каждый из n кубитов в кодовом блоке, а результирующее состояние разложено по операторам Паули, как в уравнении (7.45), тогда состояние $|\mathrm{BAD}\rangle$ (возникающее из слагаемых, в которых $\mathbf{W}$ действует по крайней мере на $t+1$ кубитов) имеет норму порядка $(\sqrt{p})^{t+1}$, а неравенство (7.58) приобретает вид

$$1 - F \leqslant O(p^{t+1}). \tag{7.65}$$

Таким образом, кодирование обеспечивает увеличение точности воспроизведения одного и того же порядка независимо от того, возникают ли некоррелированные ошибки вследствие декогерентизации или неизвестного унитарного преобразования.

Чтобы избежать путаницы, подчеркнем значение слова «некоррелированный» для ясного понимания предыдущего обсуждения. Мы рассматриваем действующую на n кубитов унитарную ошибку как «некоррелированную», если она является тензорным произведением однокубитовых унитарных преобразований, независимо от того, как могут быть связаны друг с другом унитарные преобразования, действующие на различные кубиты. Например, коррекция квантовых ошибок эффективно справляется с ошибкой, приводящей к повороту всех кубитов на угол θ вокруг общей оси. Если код может защитить от t некоррелированных ошибок, то точность

воспроизведения после восстановления равна $F = 1 - O(\theta^{2(t+1)})$. Напротив, больше трудностей вызвала бы унитарная ошибка вида $\mathbf{U}^{(n)} \sim \mathbf{1} + i\theta\mathbf{E}_{\mathrm{bad}}^{(n)}$, где $\mathbf{E}_{\mathrm{bad}}^{(n)}$ — n-кубитовый оператор Паули с весом, бóльшим t. В этом случае $|\mathrm{BAD}\rangle$ имеет норму порядка θ, а типичная точность воспроизведения после восстановления равна $F = 1 - O(\theta^2)$.

7.5. Классические линейные коды

Квантовые коды коррекции ошибок впервые были изобретены менее четырех лет тому назад,[1] но классические коды коррекции ошибок имеют гораздо более длинную историю. За последние пятьдесят лет была построена удивительно красивая и мощная теория классического кодирования. Многое из нее может быть использовано при создании КККО. Здесь мы сделаем беглый обзор лишь некоторых элементов классической теории, акцентируя наше внимание на двоичных линейных кодах.

В двоичном коде k битов кодируются двоичной строкой длины n. То есть из 2^n строк длины n мы выбираем подмножество, содержащее 2^k строк — кодовых слов; k-битовое сообщение кодируется путем отбора одного из этих 2^k кодовых слов.

В частном случае двоичного линейного кода кодовые слова образуют k-мерное замкнутое линейное подпространство C двоичного векторного пространства F_2^n. То есть побитовое исключающее ИЛИ (XOR) двух кодовых слов является другим кодовым словом. Пространство кода C натянуто на базис из k векторов $v_1, v_2, \ldots, v_k$; произвольное кодовое слово можно представить в виде линейной комбинации этих базисных векторов

$$v(\alpha_1, \alpha_2, \ldots, \alpha_k) = \sum_i \alpha_i v_i, \tag{7.66}$$

где каждое $\alpha_i \in \{0, 1\}$, а сложение выполняется по модулю 2. Можно сказать, что вектор $v(\alpha_1, \alpha_2, \ldots, \alpha_k)$ длины n кодирует k-битовое сообщение $\alpha = (\alpha_1, \ldots, \alpha_k)$.

k базисных векторов $v_1, v_2, \ldots, v_k$ можно скомпоновать в $k \times n$-матрицу

$$G = \begin{pmatrix} v_1 \\ \vdots \\ v_k \end{pmatrix}, \tag{7.67}$$

[1] Отсчитывая от 1998 года, времени опубликования английского издания этих лекций. — *Прим. ред.*

которая называется *генерирующей матрицей* кода. Тогда в матричных обозначениях уравнение (7.66) может быть переписано как

$$v(\alpha) = \alpha G; \tag{7.68}$$

действующая налево матрица G кодирует сообщение α.

Альтернативный способ описания k-мерного кодового подпространства векторного пространства F_2^n состоит в определении $n-k$ линейных условий. Существует такая $(n-k) \times n$-матрица H, что

$$Hv = 0 \tag{7.69}$$

для всех тех и только тех векторов v, которые принадлежат коду C. Эта матрица H называется матрицей контроля четности кода C. Строки матрицы H образуют $n-k$ линейно независимых векторов, а кодовым пространством является пространство *ортогональных* им векторов. Ортогональность определяется относительно побитового внутреннего произведения по модулю 2; две двоичные строки длины n ортогональны, если они «сталкиваются» («collide» — обе принимают значение 1) в четном количестве позиций. Отметим, что

$$HG^T = 0, \tag{7.70}$$

где G^T — транспонированная матрица G; строки G ортогональны строкам H.

Единственным типом ошибки классического бита является его инвертирование. Возникшую в n-битовой строке ошибку можно характеризовать n-компонентным вектором e, где единицы в e показывают позиции, в которых появились ошибки. Возмущенная ошибкой e строка v принимает вид

$$v \rightarrow v + e. \tag{7.71}$$

Ошибки можно обнаружить, применяя матрицу контроля четности. Если v является кодовым словом, то

$$H(v+e) = Hv + He = He. \tag{7.72}$$

Вектор He называется синдромом ошибки e. Обозначим через $\mathcal{E}$ множество ошибок $\{e_i\}$, которые мы хотим уметь корректировать. Исправление ошибок будет возможно, если и только если все они имеют различные синдромы. Только при этом условии можно однозначно выявить ошибку с синдромом He, а затем исправить ее, инвертируя отмеченные в e биты,

$$v + e \rightarrow (v+e) + e = v. \tag{7.73}$$

С другой стороны, если $He_1 = He_2$ при $e_1 \neq e_2$, то мы можем неправильно интерпретировать ошибку e_1 как e_2. Тогда попытка исправления произведет следующий эффект:

$$v + e_1 \rightarrow v + (e_1 + e_2) \neq v. \tag{7.74}$$

Восстановленное сообщение $v + e_1 + e_2$ принадлежит подпространству кодов, но оно отличается от исходного сообщения v; закодированная информация повреждена.

Расстояние d кода C представляет собой минимальное значение веса ненулевых векторов $v \in C$, где *вес* равен количеству единиц в строке v. Линейный код с расстоянием $d = 2t + 1$ может исправить t ошибок; код сопоставляет конкретный синдром каждой $e \in \mathcal{E}$, где $\mathcal{E}$ содержит все векторы с весом, не превышающим t. Это так, поскольку если $He_1 = He_2$, то

$$0 = He_1 + He_2 = H(e_1 + e_2) \tag{7.75}$$

и, следовательно, $e_1 + e_2 \in C$. Но если e_1 и e_2 не равны между собой и каждый имеет вес, не превышающий t, то вес $e_1 + e_2$ больше нуля, но не превышает $2t$. Но поскольку $d = 2t + 1$, то вектор $e_1 + e_2$ не может принадлежать C. Следовательно, He_1 и He_2 не могут быть равны друг другу.

Полезной конструкцией классической теории кодирования является *дуальный* (*двойственный*) *код*. Мы видели, что генерирующая $k \times n$-матрица G и $(n-k) \times n$-матрица контроля четности H кода C связаны соотношением $HG^T = 0$. Транспонируя это равенство, получим $GH^T = 0$. Таким образом, мы можем рассматривать H как генератор, а G — как матрицу контроля четности $(n-k)$-мерного кода (этот код обозначается как $C^\perp$ и называется дуальным по отношению к C). Другими словами, $C^\perp$ представляет собой ортогональное дополнение C в F_2^n. Вектор является самоортогональным, если он имеет четный вес, следовательно, возможно пересечение C и $C^\perp$. Код содержит дуальный ему код, если все его кодовые слова имеют четный вес и взаимно ортогональны. Если $n = 2k$, то возможно, что $C = C^\perp$, в этом случае говорят, что код C является *самодуальным* (или *самодвойственным*).

В следующем разделе окажется полезным тождество, связывающее код C с дуальным ему кодом $C^\perp$

$$\sum_{v \in C} (-1)^{v \cdot u} = \begin{cases} 2^k & u \in C^\perp, \\ 0 & u \notin C^\perp. \end{cases} \tag{7.76}$$

Нетривиальным содержанием этого тождества является утверждение, что сумма обращается в нуль для $u \notin C^{\perp}$. Это непосредственно вытекает из знакомого тождества

$$\sum_{v \in \{0,1\}^k} (-1)^{v \cdot w} = 0, \qquad w \neq 0, \tag{7.77}$$

где v и w — строки длины k. Мы можем представить $v \in C$ как

$$v = \alpha G, \tag{7.78}$$

где α представляет собой k-мерный вектор. Тогда

$$\sum_{v \in C} (-1)^{v \cdot u} = \sum_{\alpha \in \{0,1\}^k} (-1)^{\alpha \cdot Gu} = 0, \tag{7.79}$$

для $Gu \neq 0$. Так как G, генерирующая матрица кода C, является матрицей контроля четности кода $C^{\perp}$, то мы приходим к выводу, что сумма обращается в нуль при $u \notin C^{\perp}$.

7.6. Коды КШС

При создании квантовых кодов коррекции ошибок можно применять принципы теории классических линейных кодов. Здесь мы опишем семейство ККК0 — кодов Колдербэнка–Шора–Стина (КШС), при построении которых используется понятие дуального кода.

Пусть C_1 — классический линейный код с $(n - k_1) \times n$-матрицей контроля четности H_1, и пусть C_2 — *субкод* кода C_1 с $(n - k_2) \times n$-матрицей контроля четности H_2, где $k_2 < k_1$. Первые $n - k_1$ строк матрицы H_2 совпадают с первыми $n - k_1$ строками матрицы H_1, но в H_2 содержится $k_1 - k_2$ дополнительных линейно независимых строк; таким образом, каждое слово из C_2 содержится в C_1, но слова из C_2 подчиняются некоторым дополнительным линейным условиям.

Субкод C_2 определяет отношение эквивалентности в C_1; мы говорим, что $u, w \in C_1$ эквивалентны ($u \equiv w$), если и только если в C_2 существует v, такое, что $u = w + v$. Классы эквивалентности являются *смежными классами*, порождаемыми субкодом C_2 в C_1.[1]

КШС-код представляет собой квантовый код с $k = k_1 - k_2$, в котором каждому классу эквивалентности, определяемому элементами $v \in C_2$, соответствует кодовое слово. Каждый элемент базиса кодового подпространства может быть представлен в виде

$$|\bar{w}\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{v \in C_2} |v + w\rangle \tag{7.80}$$

[1]Или короче, *смежными классами* C_1/C_2. — *Прим. ред.*

— равновзвешенной суперпозиции всех слов смежного класса, которому принадлежит элемент w. Существует $2^{k_1-k_2}$ смежных классов и, следовательно, $2^{k_1-k_2}$ линейно независимых кодовых слов. Состояния $|\bar{w}\rangle$ очевидно нормированы и взаимно ортогональны; то есть $\langle\bar{w}|\bar{w}'\rangle = 0$, если $\bar{w}$ и $\bar{w}'$ принадлежат разным смежным классам.

Рассмотрим теперь, что произойдет с кодовым словом $|\bar{w}\rangle$, если мы применим к нему побитовое преобразование Адамара $\mathbf{H}^{(n)}$

$$\begin{aligned}\mathbf{H}^{(n)} : |\bar{w}\rangle_F \equiv \frac{1}{\sqrt{2^{k_2}}}\sum_{v\in C_2}|v+w\rangle \to \\ \to |\bar{w}\rangle_P \equiv \frac{1}{\sqrt{2^n}}\sum_u \frac{1}{\sqrt{2^{k_2}}}\sum_{v\in C_2}(-1)^{u\cdot v}(-1)^{u\cdot w}|u\rangle = \\ = \frac{1}{\sqrt{2^{n-k_2}}}\sum_{u\in C_2^\perp}(-1)^{u\cdot w}|u\rangle;\end{aligned} \tag{7.81}$$

мы получаем взвешенную фазовыми множителями когерентную суперпозицию слов, принадлежащих дуальному коду $C_2^\perp$ [на последнем этапе мы использовали тождество (7.76)]. И снова в этом последнем выражении очевидно, что кодовое слово зависит лишь от смежного класса C_2, который представляет w, — сдвиг w на элемент, принадлежащий C_2, не влияет на $(-1)^{u\cdot w}$, если u принадлежит дуальному по отношению к C_2 коду.

Теперь предположим, что код C_1 имеет расстояние d_1, а код $C_2^\perp$ — расстояние $d_2^\perp$, такие, что

$$d_1 \geqslant 2t_F + 1, \qquad d_2^\perp \geqslant 2t_P + 1. \tag{7.82}$$

Тогда нетрудно видеть, что соответствующий КШС-код может корректировать t_F инвертированных битов и t_P обращений фазы. Пусть e — двоичная строка длины n, а $\mathbf{E}_e^{\text{flip}}$ обозначает оператор Паули с $\mathbf{X}$, действующими в каждой позиции i, в которой $e_i = 1$; он действует на состояние $|v\rangle$ по правилу

$$\mathbf{E}_e^{\text{flip}} : |v\rangle \to |v+e\rangle. \tag{7.83}$$

Пусть также $\mathbf{E}_e^{\text{phase}}$ обозначает оператор Паули с $\mathbf{Z}$, действующими в каждой позиции i, в которой $e_i = 1$; его действие представляет собой

$$\mathbf{E}_e^{\text{phase}} : |v\rangle \to (-1)^{v\cdot e}|v\rangle, \tag{7.84}$$

что в базисе, повернутом преобразованием Адамара, приобретает вид

$$\mathbf{E}_e^{\text{phase}} : |u\rangle \to |u+e\rangle. \tag{7.85}$$

Теперь, в исходном базисе (F, или «flip»-базис), каждое базисное состояние КШС-кода $|\bar{w}\rangle_F$ представляет собой суперпозицию слов кода C_1. Чтобы диагностировать ошибку инвертирования бита, мы применяем к информации и служебному кубиту унитарное преобразование

$$|v\rangle \otimes |0\rangle_A \to |v\rangle \otimes |H_1 v\rangle_A, \tag{7.86}$$

а затем измеряем служебный кубит. Результат измерения $H_1 e_F$ представляет собой *синдром инвертирования бита*. Если количество инвертированных битов не превышает t_F, то из этого синдрома мы можем сделать правильный вывод о том, что инвертирования возникли в позициях, отмеченных e_F. Применяя $\mathbf{X}$ к кубитам в этих позициях, мы восстанавливаем закодированную информацию.

Для исправления фазовых ошибок мы предварительно выполняем побитовое преобразование Адамара, чтобы перейти от базиса F к базису P («phase»). В базисе P каждое базисное состояние КШС-кода $|\bar{w}\rangle_P$ представляет собой суперпозицию слов кода $C_2^\perp$. Чтобы диагностировать фазовые ошибки, мы выполняем унитарное преобразование

$$|v\rangle \otimes |0\rangle_A \to |v\rangle \otimes |G_2 v\rangle_A \tag{7.87}$$

и измеряем служебный кубит (G_2, генерирующая матрица кода C_2, одновременно является матрицей контроля четности кода $C_2^\perp$). Результат измерения $G_2 e_P$ представляет собой *синдром фазовой ошибки*. Если количество фазовых ошибок не превосходит t_P, то из этого синдрома мы можем сделать правильный вывод о том, что фазовые ошибки возникли в позициях, отмеченных e_P. Применяя $\mathbf{X}$ (в базисе P) к кубитам в этих позициях, мы восстанавливаем закодированную информацию. Наконец, мы еще раз применяем побитовое преобразование Адамара, чтобы вернуть кодовые слова в исходный базис. (Эквивалентно, мы можем исправить фазовые ошибки, применив $\mathbf{Z}$ к поврежденным кубитам, после возвращения в базис F.)

Если e_F имеет вес меньше, чем d_1, а e_P — меньше, чем $d_2^\perp$, то

$$\langle \bar{w} | \mathbf{E}_{e_P}^{\text{phase}} \mathbf{E}_{e_F}^{\text{flip}} | \bar{w}' \rangle = 0 \tag{7.88}$$

(за исключением случая $e_P = e_F = 0$). Любой оператор Паули может быть представлен в виде произведения фазового оператора и оператора инвертирования. С этой точки зрения $\mathbf{Y}$ представляет собой инвертирование бита и обращение фазы, одновременно возмущающие один и тот же кубит. Итак, расстояние d КШС-кода удовлетворяет условию

$$d \geqslant \min(d_1, d_2^\perp). \tag{7.89}$$

КШС-коды обладают особым свойством (отсутствующим у более общих ККХО): процедуру восстановления можно разбить на две отдельные операции, одна корректирует инвертирование битов, а вторая — фазовые ошибки.

Унитарные преобразования (7.86) [или (7.87)] можно осуществить, выполняя простую квантовую схему. Нужно извлечь бит синдрома, связанный с каждой из $n-k_1$ строк матрицы контроля четности H_1. Чтобы найти a-й бит синдрома, мы готовим служебный бит в состоянии $|0\rangle_{A,a}$ и для каждого значения λ с $(H_1)_{a\lambda}=1$ осуществляем вентиль CNOT со служебным битом в качестве цели и кубитом λ в блоке данных в качестве управляющего. После измерения служебный кубит показывает значение контрольного разряда четности $\sum_\lambda (H_1)_{a\lambda} v_\lambda$. Чтобы выявить ошибки инвертирования бита и фазовые ошибки, измеряются различные синдромы. Важный частный случай конструкции КШС возникает, когда код C содержит свой дуальный код $C^\perp$. Тогда мы можем выбрать $C_1=C$, а $C_2=C^\perp\subseteq C$; в обоих базисах, F и P, вычисляется матрица контроля четности кода C, чтобы определить два синдрома.

На рисунке изображена полная квантовая схема простейшего из КШС-кодов — 7-кубитового кода Стина, рассматриваемого в следующем разделе.

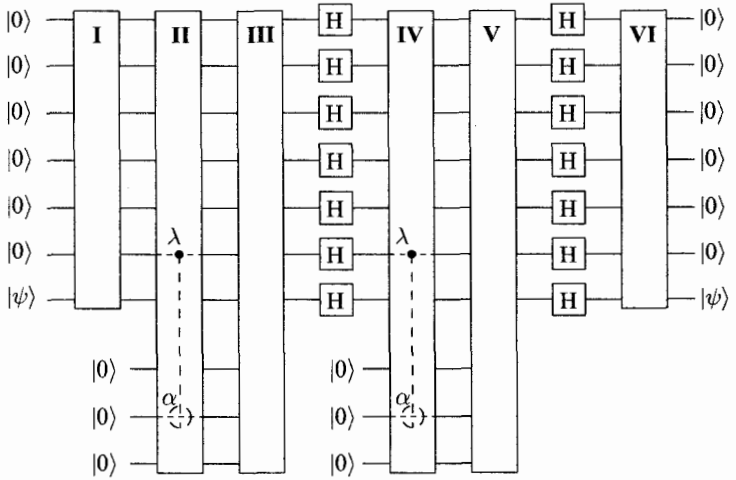
7.7. 7-кубитовый код

Простейшим из КШС-кодов является впервые сформулированный Эндрю Стином квантовый код $[[n,k,d]]=[[7,1,3]]$. Он построен по аналогии с классическим 7-битовым кодом Хэмминга.

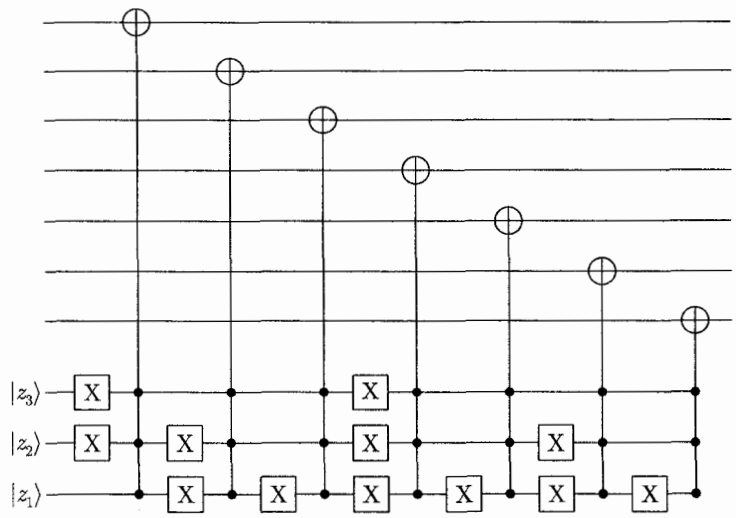
Код Хэмминга представляет собой классический код $[n,k,d]=[7,4,3]$ с 3×7-матрицей контроля четности:

$$H=\begin{pmatrix}1&0&1&0&1&0&1\\0&1&1&0&0&1&1\\0&0&0&1&1&1&1\end{pmatrix}. \tag{7.90}$$

Чтобы увидеть, что расстояние кода $d=3$, заметим сначала, что строка (1110000) с весом, равным трем, проходит контроль четности и, следовательно, принадлежит коду. Теперь нам нужно показать, что в коде нет векторов с меньшими весами. Если e_1 имеет единичный вес, то He_1 является одним из столбцов H. Но ни один из столбцов H не является тривиальным (все нули), поэтому e_1 не может принадлежать коду. Любой вектор с весом два может быть представлен как e_1+e_2, где e_1 и e_2 — различные векторы



I: Кодирование состояния $|\psi\rangle$. См. упражнение 7.6 или рис. 8 на странице 218.
II (IV): Измерение синдрома инвертирования бита (обращения фазы). См. рис. 7 на странице 214.
III (V): Коррекция ошибки инвертирования бита (обращения фазы). См. схему, изображенную ниже.
VI: Декодирование — осуществляется схемой кодирования, выполняемой в обратном направлении.



с единичным весом. Но

$$H(e_1 + e_2) = He_1 + He_2 \neq 0, \tag{7.91}$$

поскольку все столбцы матрицы H различны. Следовательно, $e_1 + e_2$ не может принадлежать коду.

Сами строки матрицы H проходят контроль четности и, следовательно, также принадлежат коду. (Вопреки интуиции, опирающейся на обычную линейную алгебру, ненулевой вектор над конечным полем F_2 может быть ортогонален самому себе.) Генерирующая матрица G кода Хэмминга может быть записана в виде

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}; \tag{7.92}$$

ее первые три строки совпадают со строками матрицы H, а в качестве четвертой строки прилагается кодовое слово (1110000) с весом три.

Дуальным по отношению к коду Хэмминга является код $[7, 3, 4]$, генерируемый матрицей H. В этом случае дуальный код на самом деле содержится в исходном коде — фактически это четный субкод кода Хэмминга, содержащий все те и только те кодовые слова, которые имеют четный вес. Нечетное кодовое слово (1110000) является представителем нетривиального смежного класса четного субкода. Для построения КШС-кода, выберем в качестве C_1 код Хэмминга, а дуальный ему четный субкод — в качестве C_2. Следовательно, $C_2^{\perp} = C_1$ также является кодом Хэмминга; мы будем использовать контроль четности Хэмминга для обнаружения ошибок инвертирования битов в базисе F и обращения фаз в базисе P.

Два ортонормированных кодовых слова этого КШС-кода, каждое из которых связано со своим смежным классом четного субкода, можно выразить в базисе F как

$$\begin{aligned} |\bar{0}\rangle_F &= \frac{1}{\sqrt{8}} \sum_{\{\text{even } v\} \in \text{Hamming}} |v\rangle, \\ |\bar{1}\rangle_F &= \frac{1}{\sqrt{8}} \sum_{\{\text{odd } v\} \in \text{Hamming}} |v\rangle. \end{aligned} \tag{7.93}$$

Так как $|\bar{0}\rangle$ и $|\bar{1}\rangle$ являются суперпозициями кодовых слов Хэмминга, то инвертирование битов можно диагностировать в этом базисе, осуществляя

контроль четности H. После адамаровского поворота базиса эти кодовые слова принимают вид

$$\mathbf{H}^{(7)}: \begin{aligned} |\bar{0}\rangle_F &\to |\bar{0}\rangle_P \equiv \frac{1}{4} \sum_{v\in\text{Hamming}} |v\rangle = \frac{1}{\sqrt{2}}(|\bar{0}\rangle_F + |\bar{1}\rangle_F), \\ |\bar{1}\rangle_F &\to |\bar{1}\rangle_P \equiv \frac{1}{4} \sum_{v\in\text{Hamming}} (-1)^{\text{wt}(v)}|v\rangle = \frac{1}{\sqrt{2}}(|\bar{0}\rangle_F - |\bar{1}\rangle_F). \end{aligned} \tag{7.94}$$

В этом базисе состояния также являются суперпозициями кодовых слов Хэмминга, так что инвертирование битов в базисе P (обращения фаз в исходном базисе) снова можно выявить с помощью контроля четности H. [Попутно отметим, что для этого кода выполнение побитового преобразования Адамара также осуществляет поворот Адамара закодированной информации; этот момент будет важен при обсуждении помехоустойчивых квантовых вычислений (см. приложение)].

Квантовый код Стина может исправить одно инвертирование бита и одно обращение фазы любого одного из семи кубитов в блоке. Но восстановление не удастся, если инвертирование бита или обращение фазы одновременно возникает в двух разных кубитах. Если e_1 и e_2 — две различные строки единичного веса, то $He_1 + He_2$ представляет собой сумму двух различных столбцов матрицы H и, следовательно, совпадает с одним из ее пяти остальных столбцов (все семь нетривиальных строк длины 3 выступают в качестве столбцов H). Поэтому существует другая строка с единичным весом, такая, что $He_1 + He_2 = He_3$, или

$$H(e_1 + e_2 + e_3) = 0; \tag{7.95}$$

таким образом, $e_1 + e_2 + e_3$ является словом кода Хэмминга с весом 3. Будем интерпретировать синдром He_3 как признак возникновения ошибки $v \to v + e_3$ и попытаемся восстановить информацию, применяя операцию $v \to v + e_3$. Тогда совокупным эффектом от двух ошибок инвертирования битов и нашей неудачной попытки восстановления будет добавление к информации $e_1 + e_2 + e_3$ (кодовое слово Хэмминга с нечетным весом), что приведет к инвертированию *закодированного* кубита

$$|\bar{0}\rangle_F \leftrightarrow |\bar{1}\rangle_F. \tag{7.96}$$

Аналогично, обращение двух фазовых множителей в базисе F эквивалентно инвертированию двух битов в базисе P, что (после неудачного восстановления) вызывает в закодированном кубите

$$|\bar{0}\rangle_P \leftrightarrow |\bar{1}\rangle_P, \tag{7.97}$$

или, что эквивариантно,

$$|\bar{0}\rangle_F \to |\bar{0}\rangle_F,$$
$$|\bar{1}\rangle_F \to -|\bar{1}\rangle_F \tag{7.98}$$

— обращение фазы закодированного кубита в базисе F. Восстановление будет успешным только в случае одновременного инвертирования одного бита и обращения одного фазового множителя (в одном или разных кубитах).

7.8. Некоторые ограничения на параметры кода

Код Шора защищает один закодированный кубит от ошибки в любом одном из девяти кубитов блока, а код Стина уменьшает размер блока с девяти до семи. Можно ли добиться лучших результатов?

7.8.1. Квантовая граница Хэмминга

Чтобы понять, насколько лучших результатов мы можем добиться, посмотрим, можно ли при данных значениях n и k найти какие-либо границы для расстояния $d = 2t + 1$ квантового кода $[[n, k, d]]$. На первых порах ограничим наше внимание *невырожденными* кодами, сопоставляющими свой синдром каждой возможной ошибке. Для данного кубита существуют три возможные линейно независимые ошибки $\mathbf{X}$, $\mathbf{Y}$ или $\mathbf{Z}$. В блоке из n кубитов имеется $\binom{n}{j}$ способов выбрать j возмущенных ошибками кубитов и три возможные ошибки для каждого из них; следовательно, общее количество возможных ошибок с весом, не превышающим t, равно

$$N(t) = \sum_{j=0}^{t} 3^j \binom{n}{j}. \tag{7.99}$$

Если закодировано k кубитов, то существует 2^k линейно независимых кодовых слов. Если все векторы $\mathbf{E}_a|\bar{j}\rangle$ линейно независимы, где $\mathbf{E}_a$ — произвольная ошибка с весом, не превышающим t, а $|\bar{j}\rangle$ — произвольный элемент базиса кодовых слов, то размерность 2^n гильбертова пространства n кубитов должна быть достаточно большой, чтобы вместить $N(t) \cdot 2^k$ независимых векторов; следовательно,

$$N(t) = \sum_{j=0}^{t} 3^j \binom{n}{j} \leqslant 2^{n-k}. \tag{7.100}$$

Этот результат называется квантовой границей Хэмминга. Аналогичная граница, но без множителя 3^j, применяется к классическим блоковым кодам, поскольку в этом случае существует только один тип ошибки (инвертирование бита), который может повредить классический бит. Подчеркнем также, что квантовая граница Хэмминга применима только в случае невырожденного кодирования, в то время как классическая граница Хэмминга применима в общем случае. Однако до сих пор не было создано ни одного вырожденного квантового кода, нарушающего квантовую границу Хэмминга (по положению на январь 1999 года).

В частном случае кода с одним закодированным кубитом ($k = 1$), исправляющим одну ошибку ($t = 1$), квантовая граница Хэмминга выглядит как

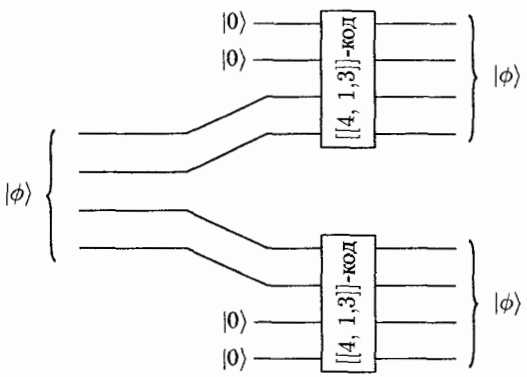
$$1 + 3n \leqslant 2^{n-1}, \tag{7.101}$$

что удовлетворяется при $n \geqslant 5$. Фактически при $n = 5$ это неравенство переходит в равенство ($1 + 15 = 16$). Невырожденный квантовый код $[[5, 1, 3]]$, если такой существует, является *совершенным*[1]. Необходимо все 32-мерное гильбертово пространство пяти кубитов, чтобы вместить все возможные действующие на кодовые слова однокубитовые ошибки — излишков пространства нет.

7.8.2. Граница невозможности клонирования

Было бы удивительно, если бы существовал *вырожденный* код $n = 4$, способный исправить одну ошибку. В действительности нетрудно увидеть, что существование такого кода невозможно. Мы уже знаем, что корректирующий t произвольно расположенных ошибок код можно также использовать для коррекции $2t$ ошибок в известных позициях. Предположим, что мы имеем некий квантовый код $[[4, 1, 3]]$. Тогда мы могли бы закодировать один кубит в 4-кубитовом блоке и разделить его на два субблока, по два кубита в каждом (см. рис. на с. 42). При добавлении к каждому из этих субблоков $|00\rangle$ исходный блок порождает двух потомков с двумя локализованными ошибками в каждом. Если бы мы могли исправить две локализованные ошибки в каждом из этих потомков, то получили бы две идентичные копии исходного блока, то есть клонировали неизвестное квантовое состояние, что невозможно. Следовательно, квантовый код $[[4, 1, 3]]$ невозможен. Таким образом, $n = 5$ представляет собой минимальный размер блока квантового кода коррекции ошибок, независимо от того, вырожден он или нет.

[1] По определению классический код называется *совершенным*, если он насыщает классическую границу Хэмминга. Классический совершенный код, исправляющий t ошибок, может исправить все ошибки с весом, не превышающим t, и не может исправить ни одной ошибки с весом, большим t. Здесь это понятие распространяется на квантовые коды. — *Прим. ред.*



Аналогичные рассуждения показывают, что код $[[n, k \geqslant 1, d]]$ возможен только при

$$n > 2(d-1). \tag{7.102}$$

7.8.3. Квантовая граница Синглтона

Сейчас мы увидим, что результат (7.102) можно усилить до

$$n - k \geqslant 2(d-1). \tag{7.103}$$

Неравенство (7.103) напоминает границу Синглтона для параметров классического кода

$$n - k \geqslant d - 1 \tag{7.104}$$

и поэтому называется «квантовой границей Синглтона». Для классического *линейного* кода неравенство (7.104) почти тривиально: код может иметь расстояние d, если только любые $d-1$ столбцов матрицы контроля четности линейно независимы. Так как столбцы имеют длину $n-k$, то линейно независимыми могут быть не более чем $n-k$ столбцов; следовательно, $d-1$ не может превосходить $n-k$. Классическая граница Синглтона справедлива и для нелинейных кодов.

Можно найти изящное доказательство квантовой границы Синглтона, использующее субаддитивность рассмотренной в разделе 5.2 энтропии фон Неймана. Для начала определим k-кубитовое вспомогательное (служебное) пространство и построим чистое состояние, в котором векторы этого служебного пространства масимально запутаны с 2^k кодовыми словами ККО

$$|\Psi\rangle_{AQ} = \frac{1}{\sqrt{2^k}} \sum |x\rangle_A |\bar{x}\rangle_Q, \tag{7.105}$$

где $\{|x\rangle_A\}$ обозначает ортонормированный базис 2^k-мерного служебного гильбертова пространства k кубитов, а $\{|\bar{x}\rangle_Q\}$ — ортонормированный базис 2^k-мерного кодового подпространства. Вычисляя след по кодовому блоку Q длины n, получим матрицу плотности служебных кубитов $\rho_A = \frac{1}{2^k}\mathbf{1}$, которой соответствует энтропия

$$S(A) = k = S(Q). \tag{7.106}$$

Теперь, если код имеет расстояние d, то может быть исправлено $d-1$ локализованных ошибок, или, как мы уже видели, ни одна наблюдаемая, действующая на $d-1$ кубитов из n, не может открыть никакой информации о закодированном состоянии. Или, что эквивалентно, наблюдаемая ничего не может сказать о состоянии служебных кубитов, закодированном в состоянии $|\Psi\rangle$.

Поскольку мы только что узнали, что $n > 2(d-1)$ (если $k \geqslant 1$), то мысленно разделим кодовый блок Q на три непересекающиеся части: множество $d-1$ кубитов $Q^{(1)}_{d-1}$, другое непересекающееся с предыдущим множество $d-1$ кубитов $Q^{(2)}_{d-1}$ и множество остальных кубитов $Q^{(3)}_{n-2(d-1)}$. Если вычислить след по состояниям кубитов из $Q^{(2)}$ и $Q^{(3)}$, то полученная в результате матрица плотности не должна содержать никаких корреляций между $Q^{(1)}$ и служебными кубитами A. Это означает, что энтропия системы $AQ^{(1)}$ аддитивна

$$S(Q^{(2)}Q^{(3)}) = S(AQ^{(1)}) = S(A) + S(Q^{(1)}). \tag{7.107}$$

Аналогично,

$$S(Q^{(1)}Q^{(3)}) = S(AQ^{(2)}) = S(A) + S(Q^{(2)}). \tag{7.108}$$

Более того, в общем случае энтропия фон Неймана субаддитивна, так что

$$\begin{aligned} S(Q^{(1)}Q^{(3)}) &\leqslant S(Q^{(1)}) + S(Q^{(3)}), \\ S(Q^{(2)}Q^{(3)}) &\leqslant S(Q^{(2)}) + S(Q^{(3)}). \end{aligned} \tag{7.109}$$

Объединяя эти неравенства с предыдущими уравнениями, получим

$$\begin{aligned} S(A) + S(Q^{(2)}) &\leqslant S(Q^{(1)}) + S(Q^{(3)}), \\ S(A) + S(Q^{(1)}) &\leqslant S(Q^{(2)}) + S(Q^{(3)}). \end{aligned} \tag{7.110}$$

Оба эти неравенства могут быть одновременно удовлетворены, если только

$$S(A) \leqslant S(Q^{(3)}). \tag{7.111}$$

Множество $Q^{(3)}$ имеет размерность $n-2(d-1)$, и его энтропия ограничена сверху этой величиной, так что

$$S(A) = k \leqslant n - 2(d-1), \tag{7.112}$$

что и представляет собой квантовую границу Синглтона.

Код $[[5,1,3]]$ насыщает эту границу, но большинству кодов с другими значениями n и k до нее далеко. Рейнс получил более сильный результат, согласно которому код $[[n,k,2t+1]]$ при $k \geqslant 1$ должен удовлетворять неравенству

$$t \leqslant \left[\frac{n+1}{6}\right], \tag{7.113}$$

где $[x]$ — «целая часть x», то есть наибольшее целое число, не превосходящее x. Таким образом, минимальная длина кода $k=1$, который может исправить $t=1,2,3,4,5$ ошибок, равна $n=5,11,17,23,29$, соответственно. Коды со всеми этими параметрами фактически построены, за исключением кода $[[23,1,9]]$.

7.9. Стабилизирующие коды

7.9.1. Общая формулировка

Мы сможем построить (невырожденный) квантовый код $[[5,1,3]]$, но для этого нам потребуется более мощная процедура построения квантовых кодов, нежели процедура КШС.

Как мы помним, чтобы установить критерий возможности исправления ошибок, оказалось весьма полезно разложить супероператор ошибок по n-кубитовым операторам Паули. Однако до сих пор никак не использовалась групповая структура множества этих операторов (произведение операторов Паули также является оператором Паули). Фактически, мы увидим, что теория групп является мощным инструментом ККК0.

Теперь нам удобнее, чтобы все операторы Паули, действующие на отдельные кубиты, были представлены вещественными матрицами, поэтому здесь символ $\mathbf{Y}$ будет обозначать антиэрмитову матрицу

$$\mathbf{Y} = \mathbf{ZX} = i\boldsymbol{\sigma}_y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \tag{7.114}$$

удовлетворяющую условию $\mathbf{Y}^2 = -\mathbf{1}$. Тогда операторы

$$\{\pm\mathbf{1}, \pm\mathbf{X}, \pm\mathbf{Y}, \pm\mathbf{Z}\} \equiv \pm\{\mathbf{1}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\} \tag{7.115}$$

являются элементами группы восьмого порядка.[1] n-кратное тензорное произведение однокубитовых операторов Паули также образует группу

$$G_n = \pm\{\mathbf{1}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}^{\otimes n}, \tag{7.116}$$

порядок которой $|G_n| = 2^{2n+1}$ (так как существует 4^n возможных тензорных произведений, а еще один множитель 2 учитывает знаки $\pm$); будем называть G_n *n-кубитовой группой Паули*. (По сути, мы будем использовать термин «группа Паули» для обозначения как самой абстрактной группы G_n, так и ее 2^n-мерного точного унитарного представления тензорными произведениями 2×2-матриц, ее единственного неприводимого представления с размерностью больше единицы.) Отметим, что G_n имеет состоящий из двух элементов центр $Z_2 = \{\pm\mathbf{1}^{\otimes n}\}$. С помощью центра как нормального делителя группы мы получаем фактор-группу $\bar{G}_n = G/Z_2$; эту группу можно также рассматривать как двоичное векторное пространство размерности 2^{2n}, этим свойством мы будем пользоваться ниже.

Группа Паули G_n (ее 2^n-мерное представление), очевидно, обладает следующими свойствами:

(i) Каждый элемент $\mathbf{M} \in G_n$ является унитарным $\mathbf{M}^{-1} = \mathbf{M}^\dagger$.

(ii) Для каждого элемента $\mathbf{M} \in G_n$, $\mathbf{M}^2 = \pm\mathbf{1} = \pm\mathbf{1}^{\otimes n}$. Более того, $\mathbf{M}^2 = +\mathbf{1}$, если количество операторов $\mathbf{Y}$ в соответствующем тензорном произведении четно, и $\mathbf{M}^2 = -\mathbf{1}$, если количество сомножителей $\mathbf{Y}$ нечетно.

(iii) Если $\mathbf{M}^2 = +\mathbf{1}$, то элемент $\mathbf{M}$ — эрмитов ($\mathbf{M} = \mathbf{M}^\dagger$); если $\mathbf{M}^2 = -\mathbf{1}$, то элемент $\mathbf{M}$ — антиэрмитов ($\mathbf{M} = -\mathbf{M}^\dagger$).

(iv) Любые два элемента $\mathbf{M}, \mathbf{N} \in G_n$ или коммутируют, или антикоммутируют: $\mathbf{MN} = \pm\mathbf{NM}$.

Мы будем использовать группу Паули, чтобы характеризовать ККК0 следующим образом: пусть S означает абелевую подгруппу n-кубитовой группы Паули G_n. Таким образом, все элементы S, действующие в $\mathcal{H}_{2^n}$, могут быть одновременно диагонализованы. Тогда связанный с S *стабилизирующий код* $\mathcal{H}_S \subseteq \mathcal{H}_{2^n}$ представляет собой общее собственное пространство всех элементов S с равным единице собственным значением.[2] То

[1] Это не группа кватернионов, а другая неабелева группа восьмого порядка — группа симметрии квадрата. Элемент четвертого порядка $\mathbf{Y}$ может рассматриваться как поворот плоскости на 90° тогда как $\mathbf{X}$ и $\mathbf{Z}$ представляют отражения в двух взаимно ортогональных осях.

[2] Стабилизирующие коды в литературе также называют *симплектическими*. — *Прим. ред.*

есть

$$|\psi\rangle \in \mathcal{H}_S, \quad \text{если} \quad \mathbf{M}|\psi\rangle = |\psi\rangle \quad \text{для всех} \quad \mathbf{M} \in S. \tag{7.117}$$

Группа S называется *стабилизатором* кода, так как она сохраняет все кодовые слова.

Группа S может быть охарактеризована ее генераторами. Это *независимые* элементы $\{\mathbf{M}_i\}$ (ни один не может быть представлен как произведение других), такие, что каждый элемент S может быть представлен как произведение элементов $\{\mathbf{M}_i\}$. Если S имеет $n-k$ генераторов, то можно показать, что кодовое пространство $\mathcal{H}_S$ имеет размерность 2^k — существует k закодированных кубитов.

Чтобы убедиться в этом, заметим прежде всего, что каждый элемент $\mathbf{M} \in S$ должен удовлетворять уравнению $\mathbf{M}^2 = +1$; если $\mathbf{M}^2 = -1$, то $\mathbf{M}$ не может иметь собственное значение $+1$. Более того, для каждого $\mathbf{M} \neq \pm 1$ из G_n, квадрат которого равен единице, собственные значения $+1$ и -1 имеют одинаковое вырождение. Это так, поскольку для каждого $\mathbf{M} \neq \pm 1$ существует антикоммутирующий с ним элемент $\mathbf{N} \in G_n$

$$\mathbf{MN} = -\mathbf{NM}; \tag{7.118}$$

следовательно, $\mathbf{M}|\psi\rangle = |\psi\rangle$, если и только если $\mathbf{M}(\mathbf{N}|\psi\rangle) = -\mathbf{N}|\psi\rangle$. Таким образом, действие унитарного оператора $\mathbf{N}$ устанавливает взаимно однозначное соответствие между собственными состояниями оператора $\mathbf{M}$ с собственными значениями $+1$ и -1. Следовательно, существует $\frac{1}{2}2^n = 2^{n-1}$ взаимно ортогональных состояний, удовлетворяющих уравнению

$$\mathbf{M}_1|\psi\rangle = |\psi\rangle, \tag{7.119}$$

где $\mathbf{M}_1$ — один из генераторов S.

Пусть теперь $\mathbf{M}_2$ — другой (коммутирующий с $\mathbf{M}_1$) элемент G_n, такой, что $\mathbf{M}_2 \neq \pm 1, \pm\mathbf{M}_1$. Мы можем найти $\mathbf{N} \in G_n$, коммутирующий с $\mathbf{M}_1$, но антикоммутирующий $\mathbf{M}_2$; следовательно, $\mathbf{N}$ сохраняет собственное пространство оператора $\mathbf{M}_1$ с собственным значением $+1$, но внутри этого пространства обменивает собственные состояния $\mathbf{M}_2$ с собственными значениями $+1$ и -1. Отсюда следует, что пространство, удовлетворяющее уравнению

$$\mathbf{M}_1|\psi\rangle = \mathbf{M}_2|\psi\rangle = |\psi\rangle, \tag{7.120}$$

имеет размерность 2^{n-2}.

Продолжая эту процедуру, заметим, что если $\mathbf{M}_j$ не зависит от $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{j-1}\}$, то существует оператор $\mathbf{N}$, коммутирующий с $\mathbf{M}_1$, $\ldots, \mathbf{M}_{j-1}$, но антикоммутирующий $\mathbf{M}_j$ (ниже мы обсудим более детально, как можно найти такой оператор $\mathbf{N}$). Следовательно, ограниченный пространством с $\mathbf{M}_1 = \mathbf{M}_2 = \ldots = \mathbf{M}_{j-1} = 1$ оператор $\mathbf{M}_j$ имеет одинаковое количество собственных векторов, соответствующих собственным значениям $+1$ и -1. Поэтому добавление еще одного генератора всегда сокращает размерность общего собственного пространства вдвое. В случае $n-k$ генераторов размерность оставшегося пространства равна $2^n(1/2)^{n-k} = 2^k$.

Язык стабилизатора полезен, поскольку он предоставляет простой способ охарактеризовать ошибки, которые код может обнаружить и исправить. Мы можем рассматривать $n-k$ генераторов стабилизатора $\mathbf{M}_1, \ldots, \mathbf{M}_{n-k}$ как *контролирующие операторы* кода, коллективные наблюдаемые, которые мы измеряем, чтобы диагностировать ошибки. Если закодированная информация не повреждена, то мы найдем $\mathbf{M}_i = 1$ для каждого генератора; но если для некоторого i окажется $\mathbf{M}_i = -1$, тогда данные ортогональны кодовому подпространству и обнаружена ошибка.

Вспомним, что супероператор ошибки может быть разложен по элементам $\mathbf{E}_a$ группы Паули. Конкретный элемент $\mathbf{E}_a$ или коммутирует, или антикоммутирует с конкретным генератором стабилизатора $\mathbf{M}$. Если $\mathbf{E}_a$ и $\mathbf{M}$ коммутируют, то

$$\mathbf{M}\mathbf{E}_a|\psi\rangle = \mathbf{E}_a\mathbf{M}|\psi\rangle = \mathbf{E}_a|\psi\rangle \tag{7.121}$$

для $|\psi\rangle \in \mathcal{H}_S$, так что ошибка сохраняет значение $\mathbf{M} = 1$. Но если $\mathbf{E}_a$ и $\mathbf{M}$ антикоммутируют, то

$$\mathbf{M}\mathbf{E}_a|\psi\rangle = -\mathbf{E}_a\mathbf{M}|\psi\rangle = -\mathbf{E}_a|\psi\rangle, \tag{7.122}$$

так что ошибка инвертирует значение $\mathbf{M}$ и, следовательно, может быть обнаружена путем измерения $\mathbf{M}$.

Для генераторов стабилизатора $\mathbf{M}_i$ и ошибок $\mathbf{E}_a$ мы можем написать

$$\mathbf{M}_i\mathbf{E}_a = (-1)^{s_{ia}}\mathbf{E}_a\mathbf{M}_i. \tag{7.123}$$

Величины s_{ia}, $i = 1, \ldots, n-k$, образуют *синдром* ошибки $\mathbf{E}_a$, поскольку при ее появлении результатом измерения $\mathbf{M}_i$ является $(-1)^{s_{ia}}$. В случае невырожденного кода величины s_{ia} различны для всех $\mathbf{E}_a \in \mathcal{E}$, так что измерение $n-k$ генераторов стабилизатора полностью диагностирует ошибку.

Найдем в более общем виде достаточное условие, которому должен удовлетворять стабилизатор, чтобы обеспечивать возможность коррекции ошибок. Вспомним, что для этого достаточно, чтобы равенство

$$\langle\psi|\mathbf{E}_a^\dagger\mathbf{E}_b|\psi\rangle = C_{ab}, \tag{7.124}$$

где C_{ab} не зависит от $|\psi\rangle$, выполнялось для любых $\mathbf{E}_a, \mathbf{E}_b \in \mathcal{E}$ и нормированного $|\psi\rangle$ из кодового подпространства. Нетрудно видеть, что это условие удовлетворяется, если для любых $\mathbf{E}_a, \mathbf{E}_b \in \mathcal{E}$ выполняется одно из нижеследующих условий:

1) $\mathbf{E}_a^\dagger\mathbf{E}_b \in S$,

2) Существует $\mathbf{M} \in S$, антикоммутирующий с $\mathbf{E}_a^\dagger\mathbf{E}_b$.

Доказательство: В случае (1) $\langle\psi|\mathbf{E}_a^\dagger\mathbf{E}_b|\psi\rangle = \langle\psi|\psi\rangle = 1$ для $|\psi\rangle \in \mathcal{H}_S$. В случае (2) предположим, что $\mathbf{M} \in S$ и $\mathbf{M}\mathbf{E}_a^\dagger\mathbf{E}_b = -\mathbf{E}_a^\dagger\mathbf{E}_b\mathbf{M}$. Тогда

$$\langle\psi|\mathbf{E}_a^\dagger\mathbf{E}_b|\psi\rangle = \langle\psi|\mathbf{E}_a^\dagger\mathbf{E}_b\mathbf{M}|\psi\rangle = -\langle\psi|\mathbf{M}\mathbf{E}_a^\dagger\mathbf{E}_b|\psi\rangle = -\langle\psi|\mathbf{E}_a^\dagger\mathbf{E}_b|\psi\rangle \tag{7.125}$$

и, следовательно, $\langle\psi|\mathbf{E}_a^\dagger\mathbf{E}_b|\psi\rangle = 0$.

Таким образом, *стабилизирующий код*, который корректирует $\{\mathcal{E}\}$, представляет собой пространство $\mathcal{H}_S$, фиксированное абелевой подгруппой S группы Паули, где любой оператор $\mathbf{E}_a^\dagger\mathbf{E}_b$ ($\mathbf{E}_a, \mathbf{E}_b \in \mathcal{E}$) удовлетворяет (1) или (2). Код является *невырожденным*, если условие (1) не выполняется ни для одного из операторов $\mathbf{E}_a^\dagger\mathbf{E}_b$.

Очевидно, мы также вполне могли бы выбрать в качестве кодового подпространства любое из 2^{n-k} совместных собственных пространств $n-k$ независимых коммутирующих элементов группы G_n. Но фактически все эти коды эквивалентны. Мы можем считать два стабилизирующих кода *эквивалентными*, если они отличаются только способом маркировки кубитов и выбором базиса для каждого однокубитового гильбертова пространства, то есть стабилизатор одного кода преобразуется в стабилизатор другого кода путем перестановки кубитов, сопровождаемой тензорным произведением унитарных однокубитовых преобразований. Если мы разделим генераторы стабилизатора на два множества $\{\mathbf{M}_1, \ldots, \mathbf{M}_j\}$ и $\{\mathbf{M}_{j+1}, \ldots, \mathbf{M}_{n-k}\}$, то существует оператор $\mathbf{N} \in G_n$, коммутирующий с каждым элементом первого множества и антикоммутирующий с каждым элементом второго множества. Применение $\mathbf{N}$ к $|\psi\rangle \in \mathcal{H}_S$ сохраняет собственные значения первого множества, одновременно инвертируя собственные значения второго. Поскольку $\mathbf{N}$ явля-

ется тензорным произведением однокубитовых унитарных преобразований, то без потери общности (с точностью до эквивалентности) все собственные значения можно выбрать равными единице. Более того, поскольку знаки минус на самом деле не имеют значения, когда стабилизатор определен, мы вполне можем сказать, что два кода эквивалентны, если, с точностью до фаз, стабилизаторы отличаются перестановкой n кубитов, а также перестановками всех индивидуальных кубитов в операторах $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$.

В процессе восстановления может произойти сбой, если существует оператор $\mathbf{E}_a^\dagger \mathbf{E}_b$, коммутирующий со стабилизатором, но не принадлежащий ему. Этот оператор сохраняет кодовое подпространство $\mathcal{H}_S$, но может действовать в нем нетривиально; таким образом, он может изменять закодированную информацию. Так как $\mathbf{E}_a|\psi\rangle$ и $\mathbf{E}_b|\psi\rangle$ имеют одинаковый синдром, мы можем неправильно принять одну ошибку $\mathbf{E}_a$ за другую — $\mathbf{E}_b$; тогда в результате действия ошибки и попытки ее исправления к информации будет применен оператор $\mathbf{E}_b^\dagger \mathbf{E}_a$, что может оказаться причиной ее повреждения.

Стабилизирующий код с расстоянием d обладает таким свойством, что любой $\mathbf{E} \in G_n$ с меньшим, чем d, весом или принадлежит стабилизатору, или антикоммутирует с его некоторым элементом. Код является невырожденным, если стабилизатор не содержит ни одного элемента с меньшим, чем d, весом. Код с расстоянием $d = 2t + 1$ может исправить t ошибок, а код с расстоянием $s + 1$ может обнаружить s ошибок или исправить s ошибок в известных позициях.

7.9.2. Симплектическая запись

Свойства стабилизирующих кодов часто лучше объясняются и выражаются на языке линейной алгебры. Стабилизатор кода S — абелева подгруппа группы Паули, имеющая порядок 2^{n-k} и состоящая из элементов, квадраты которых равны единице, — может рассматриваться как $(n-k)$-мерное замкнутое линейное подпространство пространства F_2^{2n}, самоортогональное относительно некоторого (симплектического) внутреннего произведения.

Группа $\bar{G}_n = G_n/Z_2$ изоморфна двоичному векторному пространству F_2^{2n}. Мы утверждаем это, замечая, что поскольку $\mathbf{Y} = \mathbf{ZX}$, то любой элемент $\mathbf{M}$ группы Паули (с точностью до знака $\pm$) можно представить в виде произведения операторов $\mathbf{Z}$ и $\mathbf{X}$; мы можем написать

$$\mathbf{M} = \mathbf{Z}_M \cdot \mathbf{X}_M, \tag{7.126}$$

где $\mathbf{Z}_M$ и $\mathbf{X}_M$ — тензорные произведения степеней операторов $\mathbf{Z}$ и $\mathbf{X}$ соответственно. В более явном виде оператор Паули можно записать как

$$(\alpha|\beta) \equiv \mathbf{Z}(\alpha)\mathbf{X}(\beta) = \bigotimes_{i=1}^{n} \mathbf{Z}^{\alpha_i} \cdot \bigotimes_{i=1}^{n} \mathbf{X}^{\beta_i}, \tag{7.127}$$

где α и β — двоичные строки длины n. Тогда операторы $\mathbf{Y}$ действуют в тех позициях, в которых «сталкиваются» α и β.[1] Умножение в $\bar{G}_n$ отображается на сложение в F_2^{2n}:

$$(\alpha|\beta)(\alpha'|\beta') = (-1)^{\alpha' \cdot \beta}(\alpha + \alpha'|\beta + \beta'); \tag{7.128}$$

показатель фазы $\alpha' \cdot \beta$ подсчитывает количество перестановок $\mathbf{Z}$ и $\mathbf{X}$, в процессе преобразования произведения к стандартной форме (7.127).

Из уравнения (7.128) следует, что коммутационные свойства операторов Паули можно представить в виде

$$(\alpha|\beta)(\alpha'|\beta') = (-1)^{\alpha' \cdot \beta + \alpha \cdot \beta'}(\alpha'|\beta')(\alpha|\beta). \tag{7.129}$$

Таким образом, два оператора Паули коммутируют, если и только если соответствующие векторы ортогональны относительно симплектического внутреннего произведения

$$\alpha \cdot \beta' + \alpha' \cdot \beta. \tag{7.130}$$

Отметим также, что квадрат оператора Паули равен

$$(\alpha|\beta)^2 = (-1)^{\alpha \cdot \beta}\mathbf{1}, \tag{7.131}$$

так как $\alpha \cdot \beta$ подсчитывает количество множителей $\mathbf{Y}$ в операторе; квадрат оператора Паули равен единице, если и только если

$$\alpha \cdot \beta = 0. \tag{7.132}$$

Заметим, что замкнутое подпространство, каждый элемент которого обладает этим свойством, автоматически самоортогонально, поскольку

$$\alpha \cdot \beta' + \alpha' \cdot \beta = (\alpha + \alpha') \cdot (\beta + \beta') - \alpha \cdot \beta - \alpha' \cdot \beta' = 0; \tag{7.133}$$

то есть, на языке теории групп, подгруппа G_n, квадрат каждого элемента которой равен единице, автоматически является абелевой.

[1] То есть в тех позициях, в которых в обеих двоичных строках α и β записаны единицы. При записи операторов Паули в виде (7.126), (7.127) использовано свойство тензорного произведения $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$, которое по индукции можно обобщить на произвольное количество сомножителей. — *Прим. ред.*

На языке линейной алгебры, некоторые из сделанных ранее утверждений относительно группы Паули легко проверяются путем подсчета линейных условий. Элементы являются независимыми, если линейно независимы соответствующие векторы в F_2^{2n}, так что мы можем рассматривать $n-k$ генераторов стабилизатора как базис в линейном подпространстве размерности $n-k$. Будем использовать обозначение S для линейного пространства и соответствующей абелевой группы. Тогда $S^\perp$ обозначает векторное пространство размерности $n+k$, ортогональное каждому вектору из S (относительно симплектического внутреннего произведения). Отметим, что $S^\perp$ содержит S, так как все векторы из S взаимно ортогональны. На языке теории групп, пространству $S^\perp$ соответствует нормализующая (или централизующая) группа $N(S)(\equiv S^\perp)$ группы $S \subset G_n$, то есть подгруппа группы G_n, содержащая все элементы, коммутирующие с каждым элементом S. Так как S – абелева группа, она содержится в своем собственном нормализаторе, в который входят и другие элементы (которые мы обсудим ниже). Стабилизатор кода с расстоянием d обладает свойством, согласно которому каждый элемент $(\alpha|\beta)$, вес которого $\sum_i(\alpha_i \vee \beta_i)$ меньше, чем d, или принадлежит подпространству стабилизатора S, или лежит *за пределами* ортогонального пространства $S^\perp$.

Код можно характеризовать его стабилизатором, стабилизатор — его генераторами, а $n-k$ генераторов можно представить матрицей размерности $(n-k)\times 2n$

$$H = (H_Z|H_X). \tag{7.134}$$

Здесь каждая строка представляет собой оператор Паули, записанный в виде $(\alpha|\beta)$. Синдром ошибки $\mathbf{E}_a = (\alpha_a|\beta_a)$ определяется его коммутационными свойствами с генераторами $\mathbf{M}_i = (\alpha_i'|\beta_i')$; то есть

$$s_{ia} = (\alpha_a|\beta_a)\cdot(\alpha_i'|\beta_i') = \alpha_a\cdot\beta_i' + \alpha_i'\cdot\beta_a. \tag{7.135}$$

В случае невырожденного кода каждая ошибка имеет свой собственный синдром. Если код вырожден, то возможно несколько ошибок с одним синдромом, но для их исправления мы можем применить любой из соответствующих наблюдаемому синдрому операторов $\mathbf{E}_a^\dagger$.

7.9.3. Несколько примеров стабилизирующих кодов

(а) Девятикубитовый код. Этот $[[9,1,3]]$-код имеет восемь генераторов стабилизатора, которые можно представить в виде

$$\begin{aligned} &\mathbf{Z}_1\mathbf{Z}_2, \quad \mathbf{Z}_2\mathbf{Z}_3, \quad \mathbf{Z}_4\mathbf{Z}_5, \quad \mathbf{Z}_5\mathbf{Z}_6, \quad \mathbf{Z}_7\mathbf{Z}_8, \quad \mathbf{Z}_8\mathbf{Z}_9, \\ &\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4\mathbf{X}_5\mathbf{X}_6, \quad \mathbf{X}_4\mathbf{X}_5\mathbf{X}_6\mathbf{X}_7\mathbf{X}_8\mathbf{X}_9. \end{aligned} \tag{7.136}$$

В записи (7.134) они принимают вид

$$\left(\begin{array}{cccccccccc|cccccccccc} 1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1 \end{array}\right).$$

(b) 7-кубитовый код. Этот $[[7,1,3]]$-код имеет шесть генераторов стабилизатора, которые можно записать как

$$\tilde{H} = \begin{pmatrix} H_{\text{ham}} & 0 \\ 0 & H_{\text{ham}} \end{pmatrix}, \tag{7.137}$$

где H_{ham} — 3×7-матрица контроля четности классического $[7,4,3]$-кода Хэмминга. Три контрольных оператора

$$\begin{aligned} \mathbf{M}_1 &= \mathbf{Z}_1\mathbf{Z}_3\mathbf{Z}_5\mathbf{Z}_7, \\ \mathbf{M}_2 &= \mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_6\mathbf{Z}_7, \\ \mathbf{M}_3 &= \mathbf{Z}_4\mathbf{Z}_5\mathbf{Z}_6\mathbf{Z}_7 \end{aligned} \tag{7.138}$$

обнаруживают инвертирования битов, а три контрольных оператора

$$\begin{aligned} \mathbf{M}_4 &= \mathbf{X}_1\mathbf{X}_3\mathbf{X}_5\mathbf{X}_7, \\ \mathbf{M}_5 &= \mathbf{X}_2\mathbf{X}_3\mathbf{X}_6\mathbf{X}_7, \\ \mathbf{M}_6 &= \mathbf{X}_4\mathbf{X}_5\mathbf{X}_6\mathbf{X}_7 \end{aligned} \tag{7.139}$$

обнаруживают фазовые ошибки. Пространство с $\mathbf{M}_1 = \mathbf{M}_2 = \mathbf{M}_3 = 1$ натянуто на кодовые слова, удовлетворяющие контролю четности Хэмминга. Вспоминая, что адамаровское преобразование базиса обменивает операторы $\mathbf{Z}$ и $\mathbf{X}$, мы видим, что пространство с $\mathbf{M}_4 = \mathbf{M}_5 = \mathbf{M}_6 = 1$ натянуто на кодовые слова, удовлетворяющие контролю четности Хэмминга в базисе, полученном преобразованием Адамара. Действительно, мы построили семикубитовый код, требуя, чтобы контроль четности Хэмминга удовлетворялся в обоих базисах. Генераторы коммутируют, потому что код Хэмминга содержит дуальный ему код; то есть каждая строка матрицы H_{ham} удовлетворяет контролю четности Хэмминга.

(c) КШС-коды. Вспомним, что если классический $[n, k, d]$-код C содержит дуальный ему код $C^{\perp}$, мы можем выполнить КШС-конструкцию, чтобы получить квантовый $[[n, 2k-n, d]]$-код. Стабилизатор этого кода можно записать как

$$\tilde{H} = \begin{pmatrix} H & 0 \\ 0 & H \end{pmatrix}, \tag{7.140}$$

где $H - (n-k) \times n$-матрица контроля четности кода C. Как и для семикубитового кода, стабилизаторы коммутируют, поскольку C содержит $C^{\perp}$, а кодовое подпространство натянуто на состояния, удовлетворяющие контролю четности H в F- и P-базисах. Или, что эквивалентно, кодовые слова удовлетворяют контролю четности H и инвариантны относительно

$$|v\rangle \rightarrow |v + w\rangle, \tag{7.141}$$

где $w \in C^{\perp}$.

(d) Более общие КШС-коды. Рассмотрим более общий стабилизатор, каждый генератор которого можно выбрать в виде произведения операторов $\mathbf{Z}$ $(= (\alpha|0))$ или операторов $\mathbf{X}$ $(= (0|\beta))$. Тогда генераторы имеют вид

$$\tilde{H} = \begin{pmatrix} H_Z & 0 \\ 0 & H_X \end{pmatrix}. \tag{7.142}$$

Какому условию должны удовлетворять H_X и H_Z, если $\mathbf{Z}$- и $\mathbf{X}$-генераторы коммутируют между собой? Так как операторы $\mathbf{Z}$ должны сталкиваться с операторами $\mathbf{X}$ в четном количестве позиций, мы имеем

$$H_X H_Z^T = H_Z H_X^T = 0. \tag{7.143}$$

Но это всего лишь требование того, чтобы дуальный код $C_X^{\perp}$ с матрицей контроля четности H_X содержался в коде C_Z с матрицей контроля четности H_Z. Другими словами, этот ККК0 входит в семейство КШС-кодов с

$$C_2 = C_X^{\perp} \subseteq C_1 = C_Z. \tag{7.144}$$

Итак, мы можем характеризовать КШС-коды как такие и только такие, стабилизаторы которых имеют генераторы вида (7.142).

Однако следует предостеречь: определяемый уравнением (7.142) код невырожден, если ошибки ограничены весами, меньшими, чем $d = \min(d_Z, d_X)$ (где d_Z — расстояние кода C_Z, а d_X — расстояние кода C_X). Но истинное расстояние ККК0 может превышать d. Например,

9-кубитовый код в этом обобщенном смысле является КШС-кодом. Но в этом случае классический код C_X имеет единичное расстояние, отражая, например, то, что $\mathbf{Z}_1\mathbf{Z}_2$ содержится в стабилизаторе. Тем не менее, расстояние КШС-кода $d = 3$, так как ни один оператор Паули с весом два не принадлежит $S^{\perp} \setminus S$.

7.9.4. Закодированные кубиты

Мы видели, что причиняющие беспокойство ошибки находятся в $S^{\perp} \setminus S$ — те, что коммутируют со стабилизатором, но лежат за его пределами. Эти операторы Паули интересны также и по другой причине: их можно рассматривать как «логические» операторы, действующие на закодированные данные, которые защищены кодом.

С точки зрения линейной алгебры, мы можем видеть, что нормализатор $S^{\perp}$ стабилизатора содержит $n + k$ независимых генераторов. Действительно, подпространство в $2n$-мерном пространстве векторов $(\alpha|\beta)$, содержащее векторы, ортогональные каждому из $n - k$ линейно независимых векторов, имеет размерность $2n - (n - k) = n + k$. Из $n + k$ векторов, образующих линейную оболочку этого пространства, $n - k$ можно выбрать в качестве генераторов самого стабилизатора. Оставшиеся $2k$ генераторов сохраняют кодовое пространство, так как они коммутируют со стабилизатором, но нетривиально действуют на k закодированных кубитов.

Фактически в качестве этих $2k$ операций могут быть выбраны однокубитовые операторы $\bar{\mathbf{Z}}_i$, $\bar{\mathbf{X}}_i$, $i = 1, 2, \ldots, k$, где $\bar{\mathbf{Z}}_i$, $\bar{\mathbf{X}}_i$ — операторы Паули $\mathbf{Z}$ и $\mathbf{X}$, действующие на закодированный кубит, обозначенный индексом i. Во-первых, отметим, что мы можем расширить $n - k$ генераторов стабилизатора до максимального набора n коммутирующих операторов. Добавляемые в наш набор k операторов можно обозначить как $\bar{\mathbf{Z}}_1, \ldots, \bar{\mathbf{Z}}_k$. Тогда мы можем рассматривать общие собственные состояния $\bar{\mathbf{Z}}_1, \ldots, \bar{\mathbf{Z}}_k$ (в кодовом подпространстве $\mathcal{H}_S$) как логические базисные состояния $|\bar{z}_1, \ldots, \bar{z}_k\rangle$, с $\bar{z}_j = 0$, соответствующим $\bar{\mathbf{Z}}_j = 1$, и с $\bar{z}_j = 1$, соответствующим $\bar{\mathbf{Z}}_j = -1$.

Оставшиеся k генераторов нормализатора можно выбрать взаимно коммутирующими, а также коммутирующими со стабилизатором, но тогда они не будут коммутировать с любым из операторов $\bar{\mathbf{Z}}_i$. Осуществляя процедуру ортонормирования Грамма–Шмидта, мы можем выбрать эти генераторы, обозначенные как $\bar{\mathbf{X}}_i$, чтобы диагонализовать симплектическую форму, так что

$$\bar{\mathbf{Z}}_i\bar{\mathbf{X}}_j = (-1)^{\delta_{ij}}\bar{\mathbf{X}}_j\bar{\mathbf{Z}}_i. \tag{7.145}$$

Таким образом, каждый $\bar{\mathbf{X}}_j$ обращает собственное значение соответствующего оператора $\bar{\mathbf{Z}}_j$ и, следовательно, может рассматриваться как оператор Паули $\mathbf{X}$, действующий на j-й закодированный кубит.

(a) 9-кубитовый код. Как мы обсуждали выше, в качестве логических операторов можно выбрать

$$\bar{\mathbf{X}} = \mathbf{X}_1\mathbf{X}_2\mathbf{X}_3, \qquad \bar{\mathbf{Z}} = \mathbf{Z}_1\mathbf{Z}_4\mathbf{Z}_7. \tag{7.146}$$

Они антикоммутируют между собой ($\mathbf{X}$ и $\mathbf{Z}$ сталкиваются в позиции 1), коммутируют с генераторами стабилизатора и не зависят от генераторов (ни один из элементов стабилизатора не содержит три оператора $\mathbf{X}$ или три оператора $\mathbf{Z}$).

(b) 7-кубитовый код. Мы видели, что

$$\bar{\mathbf{X}} = \mathbf{X}_1\mathbf{X}_2\mathbf{X}_3, \qquad \bar{\mathbf{Z}} = \mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3. \tag{7.147}$$

Тогда $\mathbf{X}$ добавляет нечетное кодовое слово Хэмминга, а $\mathbf{Z}$ обращает его фазу. Эти операции осуществляют инвертирование бита, соответственно, и обращение фазы в базисе $\{|0\rangle_F, |1\rangle_F\}$, определенном в уравнении (7.93).

7.10. 5-кубитовый код

Все рассмотренные до сих пор ККК0 относятся к КШС-типу — каждый генератор стабилизатора является произведением операторов $\mathbf{Z}$ или $\mathbf{X}$. Но не все стабилизирующие коды обладают этим свойством. Примером стабилизирующего кода, не относящиеся к КШС-типу, является совершенный невырожденный $[[5, 1, 3]]$-код.

Его четыре генератора стабилизатора можно представить в виде

$$\begin{aligned} \mathbf{M}_1 &= \mathbf{XZZX1}, \\ \mathbf{M}_2 &= \mathbf{1XZZX}, \\ \mathbf{M}_3 &= \mathbf{X1XZZ}, \\ \mathbf{M}_4 &= \mathbf{ZX1XZ}. \end{aligned} \tag{7.148}$$

Генераторы $\mathbf{M}_{2,3,4}$ получены из $\mathbf{M}_1$ путем выполнения циклической перестановки кубитов. (Полученный с помощью циклической перестановки кубитов пятый оператор $\mathbf{M}_5 = \mathbf{ZZX1X} = \mathbf{M}_1\mathbf{M}_2\mathbf{M}_3\mathbf{M}_4$ зависит от четырех

других.) Поскольку результатом циклической перестановки сомножителей генератора является другой генератор, код сам по себе цикличен — результатом циклической перестановки кубитов кодового слова является кодовое слово.

Очевидно, что каждый $\mathbf{M}_i$ не содержит ни одного $\mathbf{Y}$ и, следовательно, при возведении в квадрат дает $\mathbf{1}$. Для каждой пары генераторов имеет место по два столкновения между $\mathbf{X}$ и $\mathbf{Z}$, так что генераторы коммутируют. Можно быстро проверить, что каждый оператор Паули с весом единица или два антикоммутирует по крайней мере с одним генератором, так что расстояние кода равно трем.

Рассмотрим, например, существуют ли коммутирующие со всеми четырьмя генераторами операторы ошибок с носителями на первой паре кубитов. Чтобы коммутировать с $\mathbf{1X}$ в $\mathbf{M}_2$ и с $\mathbf{X1}$ в $\mathbf{M}_3$, оператор с весом два должен быть равен $\mathbf{XX}$. Но $\mathbf{XX}$ антикоммутирует с $\mathbf{XZ}$ в $\mathbf{M}_1$ и с $\mathbf{ZX}$ в $\mathbf{M}_4$. В симплектической записи стабилизатор (7.148) имеет вид

$$\tilde{H} = \left(\begin{array}{c|c} 01100 & 10010 \\ 00110 & 01001 \\ 00011 & 10100 \\ 10001 & 01010 \end{array}\right). \tag{7.149}$$

Эта матрица имеет изящную интерпретацию, так как каждый из ее столбцов можно рассматривать как синдром однокубитовой ошибки. Например, оператор однокубитового инвертирования бита $\mathbf{X}_j$ коммутирует с $\mathbf{M}_i$, если в позиции j оператор $\mathbf{M}_i$ имеет $\mathbf{1}$ или $\mathbf{X}$, и антикоммутирует, если в позиции j оператор $\mathbf{M}_i$ имеет $\mathbf{Z}$. Таким образом, таблица

	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	$\mathbf{X}_5$
$\mathbf{M}_1$	0	1	1	0	0
$\mathbf{M}_2$	0	0	1	1	0
$\mathbf{M}_3$	0	0	0	1	1
$\mathbf{M}_4$	1	0	0	0	1

составляет список результатов измерения $\mathbf{M}_{1,2,3,4}$ в случае инвертирования бита. (Например, если инвертирован первый бит, результаты измерения $\mathbf{M}_1 = \mathbf{M}_2 = \mathbf{M}_3 = 1$, $\mathbf{M}_4 = -1$ выявляют ошибку). Аналогично, правую часть $\tilde{H}$ можно рассматривать как таблицу синдромов фазовых ошибок.

	$\mathbf{Z}_1$	$\mathbf{Z}_2$	$\mathbf{Z}_3$	$\mathbf{Z}_4$	$\mathbf{Z}_5$
$\mathbf{M}_1$	1	0	0	1	0
$\mathbf{M}_2$	0	1	0	0	1
$\mathbf{M}_3$	1	0	1	0	0
$\mathbf{M}_4$	0	1	0	1	0

Так как $\mathbf{Y}$ антикоммутирует с $\mathbf{Z}$ и $\mathbf{X}$, синдром ошибки $\mathbf{Y}_i$ дает сумма i-х столбцов таблиц $\mathbf{X}$ и $\mathbf{Z}$:

	$\mathbf{Y}_1$	$\mathbf{Y}_2$	$\mathbf{Y}_3$	$\mathbf{Y}_4$	$\mathbf{Y}_5$
$\mathbf{M}_1$	1	1	1	1	0
$\mathbf{M}_2$	0	1	1	1	1
$\mathbf{M}_3$	1	0	1	1	1
$\mathbf{M}_4$	1	1	0	1	1

Путем непосредственной проверки можно убедиться в том, что все 15 столбцов таблиц синдромов $\mathbf{X}$, $\mathbf{Y}$ и $\mathbf{Z}$ различны, и, следовательно, мы вновь подтверждаем, что рассматриваемый код невырожден и корректирует одну ошибку. Действительно, код совершенен — каждая из пятнадцати нетривиальных двоичных строк длины четыре выступает в качестве столбца в одной из этих таблиц.

Благодаря свойству цикличности кода, нетрудно охарактеризовать все 15 нетривиальных элементов его стабилизатора. Помимо $\mathbf{M}_1 = \mathbf{XZZX1}$ и четырех операторов, получаемых из него путем циклических перестановок кубитов, стабилизатор включает

$$\mathbf{M}_3\mathbf{M}_4 = -\mathbf{YXXY1} \tag{7.150}$$

плюс все его циклические перестановки, а также

$$\mathbf{M}_2\mathbf{M}_5 = -\mathbf{ZYYZ1} \tag{7.151}$$

и все его циклические перестановки. Очевидно, что все элементы стабилизатора являются операторами Паули с весом четыре.

В качестве логических операторов можно выбрать

$$\bar{\mathbf{Z}} = \mathbf{ZZZZZ}, \qquad \bar{\mathbf{X}} = \mathbf{XXXXX}; \tag{7.152}$$

они коммутируют с $\mathbf{M}_{1,2,3,4}$, дают в квадрате единицу $\mathbf{1}$ и антикоммутируют между собой. Имея вес пять, они сами не содержатся в стабилизаторе. Следовательно, если нас не беспокоит разрушение закодированного состояния, то мы можем определить значение $\bar{\mathbf{Z}}$ для закодированного кубита, измеряя $\mathbf{Z}$ каждого кубита и вычисляя четность результатов. Фактически, поскольку код имеет расстояние три, существуют элементы множества $S^{\perp} \setminus S$ с весом три; альтернативные выражения для $\bar{\mathbf{Z}}$ и $\bar{\mathbf{X}}$ можно получить путем умножения на элементы стабилизатора. Например, мы можем

выбрать

$$\bar{\mathbf{Z}} = (\mathbf{ZZZZZ}) \cdot (-\mathbf{ZYYZ1}) = -\mathbf{1XX1Z} \tag{7.153}$$

(или одну из его циклических перестановок) и

$$\bar{\mathbf{X}} = (\mathbf{XXXXX}) \cdot (-\mathbf{YXXY1}) = -\mathbf{Z11ZX} \tag{7.154}$$

(или одну из его циклических перестановок). Следовательно, возможно установить значение $\bar{\mathbf{X}}$ или $\bar{\mathbf{Z}}$, измеряя $\mathbf{X}$ или $\mathbf{Z}$ только трех из пяти кубитов в блоке и вычисляя четности результатов.

Если угодно, ортонормированный базис кодового подпространства можно построить следующим образом. Начиная с любого состояния $|\psi_0\rangle$, можно получить

$$|\Psi_0\rangle = \sum_{\mathbf{M} \in S} \mathbf{M}|\psi_0\rangle. \tag{7.155}$$

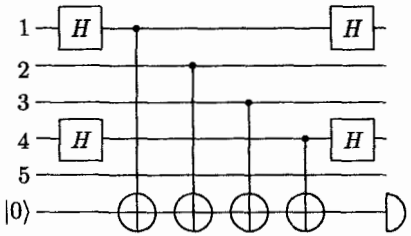
Это (ненормированное) состояние удовлетворяет условию $\mathbf{M}'|\Psi_0\rangle = |\Psi_0\rangle$ для каждого $\mathbf{M}' \in S$, так как умножение на элемент стабилизатора лишь переставляет слагаемые в сумме. Чтобы получить закодированное состояние $|\bar{0}\rangle$, отвечающее собственному значению $\bar{\mathbf{Z}} = 1$, можно начать с состояния $|00000\rangle$, которое также является собственным состоянием с $\bar{\mathbf{Z}} = 1$, но не принадлежит стабилизатору; в итоге находим (с точностью до нормировки)

$$\begin{aligned}
|\bar{0}\rangle = \sum_{\mathbf{M} \in S} \mathbf{M}|00000\rangle = \\
= |00000\rangle + (\mathbf{M}_1 + \text{циклические перестановки})|00000\rangle + \\
+ (\mathbf{M}_3\mathbf{M}_4 + \text{циклические перестановки})|00000\rangle + \\
+ (\mathbf{M}_2\mathbf{M}_5 + \text{циклические перестановки})|00000\rangle = \\
= |00000\rangle + (|10010\rangle + \text{циклические перестановки}) - \\
- (|11110\rangle + \text{циклические перестановки}) - \\
- (|01100\rangle + \text{циклические перестановки}).
\end{aligned} \tag{7.156}$$

После этого, применяя $\bar{\mathbf{X}}$ к $|\bar{0}\rangle$, то есть инвертируя все пять кубитов, можно найти

$$\begin{aligned}
|\bar{1}\rangle = \bar{\mathbf{X}}|\bar{0}\rangle = |11111\rangle + (|01101\rangle + \text{циклические перестановки}) - \\
- (|00001\rangle + \text{циклические перестановки}) - \\
- (|10011\rangle + \text{циклические перестановки}).
\end{aligned} \tag{7.157}$$

Как измеряется синдром? Возможная для выполнения измерения $\mathbf{M}_1 = \mathbf{XZZX1}$ схема изображена на рисунке.



Повороты Адамара первого и четвертого кубитов преобразуют $\mathbf{M}_1$ в тензорное произведение $\mathbf{ZZZZ1}$, а затем вентили CNOT отпечатывают значение этого оператора на служебный кубит. Заключительные повороты Адамара возвращают закодированный блок в стандартное кодовое подпространство. Схемы для измерения $\mathbf{M}_{2,3,4}$ получаются из изображенного выше путем циклической перестановки пяти кубитов в кодовом блоке.

А что можно сказать о кодировании? Мы хотим построить унитарное преобразование

$$\mathbf{U}_{\text{encode}} : |0000\rangle \otimes (a|0\rangle + b|1\rangle) \rightarrow a|\bar{0}\rangle + b|\bar{1}\rangle. \tag{7.158}$$

Мы только что видели, что $|00000\rangle$ является собственным состоянием оператора $\bar{\mathbf{Z}}$, отвечающим собственному значению $\bar{\mathbf{Z}} = 1$, а $|00001\rangle$ — собственным состоянием, отвечающим собственному значению $\bar{\mathbf{Z}} = -1$. Следовательно, (с точностью до нормировки)

$$a|\bar{0}\rangle + b|\bar{1}\rangle = \sum_{\mathbf{M}\in S} \mathbf{M}|0000\rangle \otimes (a|0\rangle + b|1\rangle). \tag{7.159}$$

Итак, нам необходимо понять, как построить схему, применяющую операцию $\sum \mathbf{M}$ к начальному состоянию.

Так как генераторы независимы, каждый элемент стабилизатора можно единственным способом представить в виде произведения генераторов и, следовательно, записать

$$\sum_{\mathbf{M}\in S} \mathbf{M} = (\mathbf{1} + \mathbf{M}_4)(\mathbf{1} + \mathbf{M}_3)(\mathbf{1} + \mathbf{M}_2)(\mathbf{1} + \mathbf{M}_1). \tag{7.160}$$

Теперь, чтобы двигаться дальше, удобно представить стабилизатор в альтернативной форме. Отметим, что, не изменяя стабилизатор, генератор $\mathbf{M}_i$ можно заменить на $\mathbf{M}_i\mathbf{M}_j$. Эта замена эквивалентна добавлению j-й строки к i-й строке матрицы $\tilde{H}$. Используя подобные операции со строками, можно выполнить процедуру Гаусса в матрице H_X размерности 4×5 и,

таким образом, получить новое представление для стабилизатора

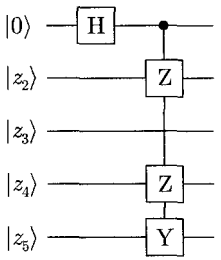
$$\tilde{H} = \left(\begin{array}{c|c} 11011 & 10001 \\ 00110 & 01001 \\ 11000 & 00101 \\ 10111 & 00011 \end{array}\right) \tag{7.161}$$

или

$$\begin{aligned} \mathbf{M}_1 &= -\mathbf{YZ1ZY}, \\ \mathbf{M}_2 &= +\mathbf{1XZZX}, \\ \mathbf{M}_3 &= +\mathbf{ZZX1X}, \\ \mathbf{M}_4 &= -\mathbf{Z1ZYY}. \end{aligned} \tag{7.162}$$

В таком виде $\mathbf{M}_i$ применяет $\mathbf{X}$ (инвертирование) только к i-у и пятому кубиту в блоке.[1]

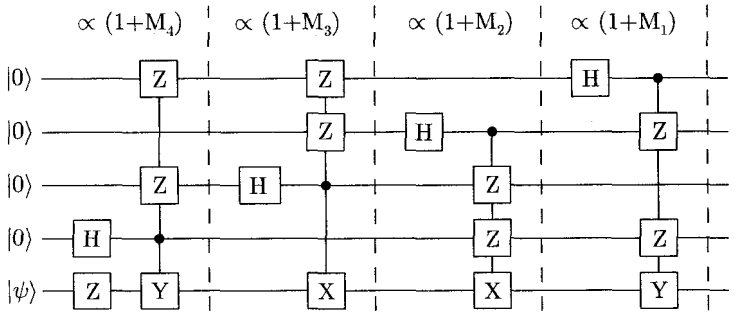
Выбирая стабилизатор в таком виде, мы можем применить $\frac{1}{\sqrt{2}}(\mathbf{1}+\mathbf{M}_1)$ к состоянию $|0, z_2, z_3, z_4, z_5\rangle$, выполняя схему



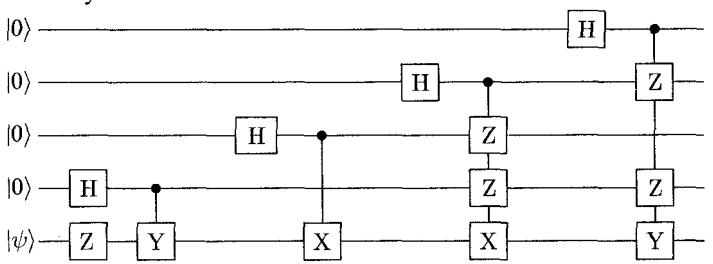
Преобразование Адамара готовит состояние $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Если первый кубит находится в состоянии $|0\rangle$, то другие операции ничего не делают, то есть результатом действия этой схемы является тождественное преобразование **11111**. Но если после адамаровского поворота первый кубит оказывается в состоянии $|1\rangle$, то остальные вентили этой схемы применяют **1Z1ZY**. В этом случае результатом является применение операции $\mathbf{M}_1 = -\mathbf{YZ1ZY}$. Можно построить подобные схемы, применяющие $\frac{1}{\sqrt{2}}(\mathbf{1}+\mathbf{M}_2)$ к состоянию $|z_1, 0, z_3, z_4, z_5\rangle$, и так далее. Кроме вентилей Адамара, каждая из этих схем подвергает действию операций $\mathbf{Z}$ и контролируемого $\mathbf{Z}$ от одного до четырех кубитов; эти кубиты никогда не инвертируют-

[1] В оригинале были пропущены знаки минус в выражениях для генераторов $\mathbf{M}_1$ и $\mathbf{M}_4$. Именно учет *правильных* знаков генераторов (7.162) приводит к *правильным* относительным знакам в выражениях для базисных векторов (7.156), (7.157). Это исправление учтено в следующих ниже рисунках и пояснениях к ним в тексте данного раздела. — *Прим. ред.*

ся. (Это убеждает нас в том, что мы выполнили процедуру Гаусса в матрице H_X.) Таким образом, можно сконструировать следующую кодирующую схему:



Более того, каждый действующий на $|0\rangle$ вентиль $\mathbf{Z}$ можно заменить на единичный, следовательно, эту схему можно упростить, исключив все такие вентили и получив



Эту процедуру можно обобщить для построения кодирующих схем для любых стабилизирующих кодов.[1]

Поскольку кодирующее преобразование унитарно, для декодирования можно использовать его сопряжение. А так как квадрат каждого вентиля равен $\pm\mathbf{1}$, то декодирующая схема представляет собой ту же самую кодирующую схему, лишь работающую в обратном направлении.

7.11. Распределение квантового секрета

Код $[[5, 1, 3]]$ является прекрасной иллюстрацией возможного применения корректирующих ошибки квантовых кодов.[2]

[1] *Естественно, что вид кодирующей схемы зависит от выбора генераторов стабилизатора. Альтернативную кодирующую схему для 5-кубитового кода (а также для других известных кодов) можно найти на сайте* http://iaks-www.ira.uka.de/home/grassl/QECC/ — *Прим. ред.*

[2] R. Cleve, D. Gottesman, and H.-K. Lo, *How to Share a Quantum Secret*, Phys. Rev. Lett. **83**, 648–651 (1999); quant-ph/9901025.

Предположим, что некоторую совершенно секретную информацию необходимо доверить n партнерам. Так как никому из них нельзя доверять полностью, секрет делится на n частей, так что каждый партнер имеет доступ только к своей части и не может узнать о секрете в целом. Но если достаточное количество партнеров соберутся и объединят свои части, то они смогут расшифровать секрет или какую-то его часть.

В частности, пороговая (m, n)-схема имеет такое свойство, что для реконструкции всей секретной информации достаточно m частей. Но из $m-1$ частей невозможно извлечь никакой информации. (Это называется пороговой схемой, потому что при собранных воедино $1, 2, 3, \ldots, m-1$ частях узнать ничего нельзя, но следующая часть позволяет переступить порог и раскрыть всю информацию.)

Следует различать два вида секретов: классический секрет представляет собой *a priori* неизвестную строку битов, в то время как квантовым секретом является *a priori* неизвестное квантовое состояние. Секреты каждого типа можно поделить. В частности, мы можем распределить классический секрет между несколькими партнерами, выбрав одно из ансамбля взаимно ортогональных (запутанных) квантовых состояний и распределив это состояние между партнерами.

Например, нетрудно видеть, что код $[[5, 1, 3]]$ может использоваться в пороговой $(3, 5)$-схеме, где разделенная информация является классической. Один классический бит кодируется одним из двух ортогональных состояний $|\bar{0}\rangle$ или $|\bar{1}\rangle$, а затем пять кубитов распределяются между пятью партнерами. Как мы уже видели, если объединятся любые два партнера, то матрицей плотности $\boldsymbol{\rho}$ их двух кубитов будет

$$\boldsymbol{\rho}^{(2)} = \frac{1}{4}\mathbf{1} \tag{7.163}$$

(поскольку код невырожден). Следовательно, из любого измерения их двух кубитов они ничего не узнают о квантовом состоянии. Но мы также видели, что код $[[5, 1, 3]]$ может исправить две локализованных ошибки или два стирания. Когда объединятся любые три участника, они могут исправить две ошибки (или восстановить два недостающих кубита) и полностью восстановить закодированное состояние $|\bar{0}\rangle$ или $|\bar{1}\rangle$.

Ясно, что с помощью аналогичной процедуры можно поделить один кубит квантовой информации — код $[[5, 1, 3]]$ также является основой квантовой пороговой ((3,5))-схемы (мы используем обозначение $((m, n))$, если поделена квантовая информация, и (m, n), если поделена классическая информация). Как эту схему деления квантового секрета распространить на большее количество кубитов? Допустим, мы приготовили чистое n-кубито-

вое состояние $|\psi\rangle$. Может ли оно быть использовано в пороговой $((m,n))$-схеме?

Нам известно, что для реконструкции состояния должно быть достаточно m кубитов; следовательно, можно восстановить $n-m$ удалений. Из общего критерия коррекции ошибок следует, что математическое ожидание любой наблюдаемой с весом, не превышающим $n-m$, не должно зависеть от состояния $|\psi\rangle$

$$\langle\psi|\mathbf{E}|\psi\rangle \quad \text{не зависит от} \quad |\psi\rangle, \quad \text{если} \quad \mathrm{wt}(\mathbf{E}) \leqslant n-m. \tag{7.164}$$

Таким образом, если m партнеров имеют всю информацию, то другие $n-m$ партнеров не имеют никакой информации. Это справедливо, поскольку квантовую информацию нельзя клонировать.

С другой стороны, мы знаем, что $m-1$ частей ничего не откроют, или что

$$\langle\psi|\mathbf{E}|\psi\rangle \quad \text{не зависит от} \quad |\psi\rangle, \quad \text{если} \quad \mathrm{wt}(\mathbf{E}) \leqslant m-1. \tag{7.165}$$

Отсюда следует, что можно восстановить $m-1$ стирание, или что другие $n-m+1$ партнеров располагают всей информацией.

Из этих двух высказываний мы получаем два неравенства

$$\begin{aligned} n-m < m &\Rightarrow n < 2m, \\ m-1 < n-m+1 &\Rightarrow n > 2m-2. \end{aligned} \tag{7.166}$$

Отсюда следует, что в квантовой пороговой $((m,n))$-схеме чистого состояния, в которой каждый партнер имеет один кубит,

$$n = 2m-1. \tag{7.167}$$

Другими словами, порог будет достигнут, когда количество наличных кубитов превысит половину всех n кубитов.

Таким образом, если каждая часть представляет собой кубит, то квантовая пороговая схема чистого состояния представляет собой квантовый код $[[2m-1,k,m]]$ с $k \geqslant 1$. Но в действительности коды $[[3,1,2]]$ и $[[7,1,4]]$ не существуют, а из границы Рейнса следует, что не существуют коды с $m > 3$. Следовательно, код $[[5,1,3]]$ представляет собой единственную квантовую пороговую схему.

Здесь следует сделать несколько оговорок. Во-первых, ограничение $n = 2m-1$ остается справедливым, даже если каждая часть является q-мерной системой, а не кубитом. Но в случае $q > 2$ можно построить различные коды:

$$[[2m-1,1,k]]_q \tag{7.168}$$

(см., например, упражнения).

Во-вторых, поделенная информация может представлять собой смешанное состояние (в котором закодировано чистое состояние). Например, если мы отбрасываем один кубит из 5-кубитового блока, мы получаем $((3,4))$-схему. Вновь, как только у нас появляется три кубита, мы сможем восстановить два стертых (то есть недостающих), один из которых находится в руках другого партнера, а второй — только что был нами же выброшен.

Наконец, мы предположили, что поделенная информация является квантовой. Но если вместо этого мы делим только классическую информацию, тогда условия восстановления стертых кубитов становятся менее строгими. Например, пару Белла можно рассматривать как вид пороговой $(2,2)$-схемы для двух битов классической информации, которая закодирована выбором одного из четырех взаимно ортогональных состояний $|\phi^{\pm}\rangle$, $|\psi^{\pm}\rangle$. Располагающий одним из двух кубитов партнер не может получить доступ к этой классической информации. Но эта схема не подходит для распределения квантового секрета, поскольку линейные комбинации этих состояний Белла *не обладают* тем свойством, что $\rho = \frac{1}{2}1$ после вычисления следа по состояниям одного из двух кубитов.

7.12. Некоторые другие стабилизирующие коды

7.12.1. Код $[[6,0,4]]$

При $k = 0$ квантовый код имеет одномерное кодовое подпространство, то есть существует только одно закодированное состояние. Код нельзя использовать для хранения неизвестной квантовой информации; тем не менее, коды с $k = 0$ могут обладать интересными свойствами. Так как они могут обнаруживать и диагностировать ошибки, они могут быть полезными для изучения корреляций в декогерентизации, вызванной взаимодействием с окружением.

Если $k = 0$, то S и $S^{\perp}$ совпадают — оператор Паули, коммутирующий со всеми элементами стабилизатора, должен принадлежать этому стабилизатору. В этом случае расстояние d определяется как минимальный вес любого принадлежащего стабилизатору оператора Паули. Таким образом, код с расстоянием d может «обнаружить $d-1$ ошибок»; то есть если любой оператор Паули с весом, меньшим d, действует на кодовое состояние, то результат будет ортогонален этому состоянию.

Закодированное состояние кода $[[6,0,4]]$ (ассоциированного с кодом $[[5,1,3]]$) можно представить в виде

$$|0\rangle \otimes |\bar{0}\rangle + |1\rangle \otimes |\bar{1}\rangle, \tag{7.169}$$

где $|\bar{0}\rangle$ и $|\bar{1}\rangle$ представляют собой собственные состояния оператора $\bar{\mathbf{Z}}$ кода $[[5,1,3]]$. Вы можете проверить, что этот код имеет расстояние $d=4$ (см. упражнение 7.3).

Код $[[6,0,4]]$ интересен тем, что его кодовое состояние максимально запутано. Мы можем выбрать любые три кубита из шести. Матрица плотности $\boldsymbol{\rho}^{(3)}$ этих трех кубитов, полученная путем вычисления следа по состояниям трех остальных, совершенно случайна, $\boldsymbol{\rho}^{(3)}=\frac{1}{8}\mathbf{1}$. В этом смысле, кодовое состояние $[[6,0,4]]$ является естественным многочастичным аналогом двухкубитовых состояний Белла. Оно «гораздо сильнее запутано», нежели шестикубитовое кот-состояние $\frac{1}{\sqrt{2}}(|000000\rangle+|111111\rangle)$. Если мы измерим в базисе $\{|0\rangle,|1\rangle\}$ любой один из шести кубитов кот-состояния, мы узнаем все о приготовленном состоянии оставшихся пяти кубитов. Но мы можем измерить по своему усмотрению любую наблюдаемую, действующую на любые *три* кубита в состоянии $[[6,0,4]]$, и ничего не узнаем относительно состояния оставшихся трех кубитов, которое по-прежнему описывается матрицей плотности $\boldsymbol{\rho}^{(3)}=\frac{1}{8}\mathbf{1}$.

Код $[[6,0,4]]$ тем более интересен, что, оказывается (но не так просто доказывается), не существует его обобщения на большее количество кубитов, то есть не существует $[[2n,0,n+1]]$ двоичных квантовых кодов для $n>3$. Однако в упражнениях вы увидите, что существуют другие, недвоичные, максимально запутанные состояния, которые можно построить.

7.12.2. Детектирующие ошибки $[[2m,2m-2,2]]$-коды

Состояние Белла $|\phi^+\rangle=\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$ представляет собой код $[[2,0,2]]$ с генераторами стабилизатора

$$\mathbf{ZZ}, \qquad \mathbf{XX}. \tag{7.170}$$

Этот код имеет расстояние два, поскольку не существует операторов Паули с единичным весом, коммутирующих с обоими генераторами (ни один из $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ одновременно не коммутирует с $\mathbf{X}$ и $\mathbf{Z}$). Соответственно, инвертирование бита ($\mathbf{X}$), обращение фазы ($\mathbf{Z}$) или обе эти ошибки ($\mathbf{Y}$), действующие на любой кубит в $|\phi^+\rangle$, преобразуют его в ортогональное состояние (одно из состояний Белла $|\phi^-\rangle$, $|\psi^+\rangle$, $|\psi^-\rangle$).

Единственный способ обобщить состояния Белла на большее количество кубитов — рассмотреть код $n=4$, $k=2$ с генераторами стабилизатора

$$\mathbf{ZZZZ}, \qquad \mathbf{XXXX}. \tag{7.171}$$

Это код с расстоянием $d = 2$ по той же причине, что и предыдущий. Кодовое подпространство натянуто на четные состояния (**ZZZZ**), инвариантные относительно одновременного инвертирования всех четырех кубитов (**XXXX**). Базис представляет собой

$$\begin{aligned} &|0000\rangle + |1111\rangle, \\ &|0011\rangle + |1100\rangle, \\ &|0101\rangle + |1010\rangle, \\ &|0110\rangle + |1001\rangle. \end{aligned} \tag{7.172}$$

Очевидно, что действующая на любой кубит ошибка **X** или **Z** преобразует каждое из этих состояний в ортогональное кодовому подпространству состояние; таким образом, можно обнаружить любую однокубитовую ошибку.

Дальнейшим обобщением является код $[[2m, 2m-2, 2]]$ с генераторами стабилизатора

$$\mathbf{ZZ}\ldots\mathbf{Z}, \quad \mathbf{XX}\ldots\mathbf{X} \tag{7.173}$$

(длина должна быть четной, чтобы генераторы коммутировали между собой). Кодовое подпространство натянуто на 2^{n-2} хорошо знакомых нам кот-состояний

$$\frac{1}{\sqrt{2}}(|x\rangle + |\neg x\rangle), \tag{7.174}$$

где x — строка длины $n = 2m$ и четного веса.

7.12.3. Код [[8, 3, 3]]

Как уже отмечалось при обсуждении кода $[[5, 1, 3]]$, стабилизирующий код с генераторами

$$\tilde{H} = (H_Z|H_X) \tag{7.175}$$

может исправить одну ошибку, если: (1) столбцы матрицы $\tilde{H}$ различны (свой синдром для каждой ошибки **X** и **Z**); (2) любая сумма столбца матрицы H_Z с соответствующим столбцом матрицы H_X отличается от любого столбца $\tilde{H}$ и от всех других таких сумм (каждую ошибку **Y** можно отличить от всех остальных однокубитовых ошибок).

Нетрудно построить матрицу H размерности 5×16 с этими свойствами и, таким образом, получить стабилизатор кода $[[8, 3, 3]]$; выберем

$$\tilde{H} = \left(\begin{array}{c|c} H & H^\sigma \\ 11111111 & 00000000 \\ 00000000 & 11111111 \end{array}\right). \tag{7.176}$$

Здесь H представляет собой матрицу размерности 3×8

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}, \tag{7.177}$$

столбцами которой являются все возможные различные двоичные последовательности длины три, а H^σ получается из H с помощью соответствующей перестановки столбцов. Эта перестановка выбирается таким образом, чтобы были различны все восемь сумм столбцов матрицы H с соответствующими столбцами H^σ. С помощью непосредственной проверки можно убедиться, что подходящим выбором служит

$$H^\sigma = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}, \tag{7.178}$$

тогда суммы столбцов равны

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{7.179}$$

Две последних строки матрицы H служат для того, чтобы отличать каждый синдром $\mathbf{X}$ от каждого синдрома $\mathbf{Y}$ или $\mathbf{Z}$, а вышеупомянутое свойство матрицы H^σ гарантирует, что все синдромы $\mathbf{Y}$ различны. Следовательно, мы построили код длины восемь с $k = 8 - 5 = 3$, который может исправить одну ошибку. Собственно, это простейший в бесконечном классе кодов $[[2^m, 2^m - m - 2, 3]]$ с $m \geqslant 3$, построенных Готтесманом.

Квантовый код $[[8, 3, 3]]$, который мы только что описали, является, так сказать, кузеном «расширенного кода Хэмминга», самодуального классического кода $[8, 4, 4]$, полученного из дуального коду Хэмминга $[7, 3, 4]$-кода путем добавления дополнительного бита четности. Его матрица контроля четности (которая также является его генерирующей матрицей) имеет вид

$$H_{\text{EH}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{7.180}$$

Эта матрица H_{EH} обладает тем свойством, что различны не только все ее восемь столбцов, но от них также отличается и *сумма* любых двух столбцов; действительно, четвертым битом суммы двух столбцов является нуль, а не единица.

7.13. Коды над GF(4)

Мы построили код $[[5,1,3]]$, угадав генераторы стабилизатора и проверив, что $d=3$. Существует ли более систематический метод?

На самом деле да. Наше подозрение, что код $[[5,1,3]]$ может существовать, возникло из наблюдения, что его параметры насыщают квантовое неравенство упаковки сфер для кодов с $t=1$

$$1+3n=2^{n-k}, \tag{7.181}$$

($16=16$ при $n=5$ и $k=1$). Для теоретика в области кодирования это уравнение может показаться знакомым.

Помимо двоичных кодов, на которых до сих пор мы концентрировали внимание, классические коды можно также построить из строк (длины n) символов, принимающих значения не в $\{0,1\}$, а в конечном поле $\mathrm{GF}(q)$, содержащем q элементов. Такие конечные поля существуют для любого $q=p^m$, где p — простое число. (GF представляет собой аббревиатуру для «Galois Field» — поле Галуа, названное так в честь его первооткрывателя.)

Для таких недвоичных кодов можно смоделировать ошибку как добавление элемента поля, циклический сдвиг q символов. Тогда всего будет $q-1$ нетривиальных ошибок. Весом вектора в $\mathrm{GF}(q)^n$ является количество его ненулевых элементов, а расстояние между двумя векторами представляет собой вес их разности (количество несовпадающих элементов). Классический код $[n,k,d]_q$ состоит из q^k кодовых слов в $\mathrm{GF}(q)^n$, где минимальное расстояние между парами строк равно d. Граница упаковки сфер, которая должна быть насыщена, для того чтобы мог существовать код $[n,k,d]_q$, при $d=3$ имеет вид

$$1+(q+1)n \leqslant q^{n-k}. \tag{7.182}$$

Насыщающие эту границу при $q=2$ совершенные двоичные коды Хэмминга с параметрами

$$n=2^m-1, \quad k=n-m \tag{7.183}$$

допускают обобщение на любое $\mathrm{GF}(q)$; совершенные коды Хэмминга над $\mathrm{GF}(q)$ можно построить при

$$n=\frac{q^m-1}{q-1}, \quad k=n-m. \tag{7.184}$$

Квантовый код $[[5,1,3]]$ происходит от классического кода Хэмминга $[5,3,3]_4$ (случай $q=4$ и $m=2$).

Что общего между классическими кодами над GF(4) и двоичными квантовыми стабилизирующими кодами? Родство возникает, потому что стабилизатору можно сопоставить замкнутое относительно сложения множество векторов над GF(4).

Поле GF(4) имеет четыре элемента, которые можно обозначить как $0, 1, \omega, \bar{\omega}$, где

$$1+1=\omega+\omega=\bar{\omega}+\bar{\omega}=0,$$
$$1+\omega=\bar{\omega} \tag{7.185}$$

и $\omega^2=\bar{\omega}$, $\omega\bar{\omega}=1$. Следовательно, аддитивная структура GF(4) соответствует мультипликативной структуре операторов группы Паули $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$. Действительно, двоичная строка $(\alpha|\beta)$ длины $2n$, которую мы использовали для обозначения элемента группы Паули, может эквивалентно рассматриваться как вектор длины n в $\mathrm{GF}(4)^n$

$$(\alpha|\beta) \leftrightarrow \alpha+\beta\omega. \tag{7.186}$$

Стабилизатор с 2^{n-k} элементами можно рассматривать как субкод кода GF(4), замкнутый относительно сложения и содержащий 2^{n-k} кодовых слов.

Отметим, что код не должен быть векторным пространством над GF(4), поскольку от него не требуется замкнутость относительно умножения на скаляр принадлежащий GF(4). В частном случае, когда код является векторным пространством, он называется *линейным* кодом.

О кодах над GF(4) известно много, поэтому эта связь открыла возможность теоретикам в области (классического) кодирования построить множество квантовых кодов коррекции ошибок.[1] Однако не каждый субкод $\mathrm{GF}(4)^n$ связан с квантовым кодом; мы до сих пор не выдвинули требование, чтобы стабилизатор был абелев — векторы $(\alpha|\beta)$, образующие линейную оболочку кода, должны быть взаимно ортогональны относительно симплектического внутреннего произведения

$$\alpha\cdot\beta'+\alpha'\cdot\beta. \tag{7.187}$$

Это условие ортогональности может выглядеть странным для теоретика в области кодирования, которому более привычно определение внутреннего произведения двух векторов в $\mathrm{GF}(4)^n$ как элемента поля GF(4), заданного соотношением

$$v*u=\bar{v}_1u_1+\ldots+\bar{v}_nu_n, \tag{7.188}$$

[1] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, *Quantum error correction via codes over* GF(4), IEEE Transact. on Inform. Theor., **44**, 1369 (1998); quant-ph/9608006.

где сопряжение, обозначенное черточкой, меняет местами ω и $\bar{\omega}$. Если это «эрмитово» внутреннее $*$-произведение двух векторов v и u равно

$$v * u = a + b\omega \in \mathrm{GF}(4), \tag{7.189}$$

то наше симплектическое внутреннее произведение равно

$$v \cdot u = b. \tag{7.190}$$

Следовательно, обращение в нуль симплектического внутреннего произведения является более слабым условием, чем обращение в нуль эрмитова внутреннего произведения. В самом деле, в частном случае *линейного* кода самоортогональность относительно эрмитова внутреннего произведения фактически эквивалентна самоортогональности относительно симплектического внутреннего произведения. Отметим, что если $v * u = a + b\omega$, то ортогональность относительно симплектического внутреннего произведения требует $b = 0$. Но если u принадлежит линейному коду, то и $\bar{\omega}u$ тоже, где

$$v * (\bar{\omega}u) = b + a\bar{\omega}, \tag{7.191}$$

так что

$$v \cdot (\bar{\omega}u) = a. \tag{7.192}$$

Мы видим, что если v и u принадлежат линейному $\mathrm{GF}(4)$-коду и ортогональны относительно симплектического внутреннего произведения, то они также ортогональны относительно эрмитова внутреннего произведения. Тогда мы делаем вывод, что линейный $\mathrm{GF}(4)$-код определяет квантовый стабилизирующий код, если и только если этот код самоортогонален относительно эрмитова внутреннего произведения. Классические коды с такими свойствами очень хорошо изучены.[1].

В частности, рассмотрим снова код Хэмминга $[5, 3, 3]_4$. Его матрицу контроля четности (в нетрадиционном представлении) можно представить в виде

$$H = \begin{pmatrix} 1 & \omega & \omega & 1 & 0 \\ 0 & 1 & \omega & \omega & 1 \end{pmatrix}, \tag{7.193}$$

что также является генерирующей матрицей дуального ему, линейного самоортогонального кода $[5, 2, 4]_4$. По сути, этот $[5, 2, 4]_4$ код с $4^2 = 16$

[1] См., например, E. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North Holland Publishing Company, Amsterdam, New York, Oxford (1977); [перевод: Ф. Дж. МакВильямс, Н. Дж. А. Слоэн, *Теория кодов, исправляющих ошибки*. — М.: Связь, 1979. — *Прим. ред.*]

кодовыми словами в точности является стабилизатором квантового кода $[[5,1,3]]$. Отождествляя $1 \equiv \mathbf{X}$, $\omega \equiv \mathbf{Z}$, мы принимаем две строки матрицы H в качестве генераторов стабилизатора $\mathbf{M}_1$, $\mathbf{M}_2$. Код, дуальный коду Хэмминга, является линейным, поэтому линейные комбинации строк принадлежат коду. Складывая строки и умножая результат на ω, получаем

$$\omega(1,\bar{\omega},0,\bar{\omega},1) = (\omega,1,0,1,\omega), \tag{7.194}$$

что представляет собой $\mathbf{M}_4$. А если к $\mathbf{M}_4$ мы добавим $\mathbf{M}_2$ и умножим на $\bar{\omega}$, то найдем

$$\bar{\omega}(\omega,0,\omega,\bar{\omega},\bar{\omega}) = (1,0,1,\omega,\omega), \tag{7.195}$$

что представляет собой $\mathbf{M}_3$.

Код $[[5,1,3]]$ является одним из примеров достаточно общей конструкции. Рассмотрим субкод C на $\mathrm{GF}(4)^n$, который является аддитивным (замкнутым относительно сложения) и самоортогональным (содержащимся в дуальном ему коде) относительно симплектического внутреннего произведения. Этот $\mathrm{GF}(4)$-код может быть отождествлен со стабилизатором двоичного ККК0 длины n. Если $\mathrm{GF}(4)$-код содержит 2^{n-k} кодовых слов, то ККК0 имеет k закодированных кубитов. Расстоянием d ККК0 является минимальный вес вектора из $C^{\perp} \setminus C$.

Другим примером самоортогонального линейного $\mathrm{GF}(4)$-кода является дуальный коду Хэмминга $m = 3$ с

$$n = \frac{1}{3}(4^3 - 1) = 21. \tag{7.196}$$

Код Хэмминга имеет 4^{n-m} кодовых слов, а дуальный ему код — $4^m = 2^6$ кодовых слов. Мы непосредственно получаем ККК0 с параметрами

$$[[21,15,3]], \tag{7.197}$$

который может исправить одну ошибку.

7.14. Хорошие квантовые коды

Семейство кодов $[[n,k,d]]$ является *хорошим*, если оно содержит коды, чья «скорость» $R = k/n$ и «вероятность возникновения ошибки» $p = t/n$ (где $t = (d-1)/2$) стремятся к ненулевым пределам при $n \to \infty$. Мы можем использовать формализм стабилизатора, чтобы доказать «квантовую границу Гилберта–Варшамова», которая демонстрирует существование хороших квантовых кодов. В сущности, хорошие коды можно выбрать невырожденными.

Мы дадим только набросок доказательства, не выполняя точно требуемые вычисления. Пусть $\mathcal{E} = \{\mathbf{E}_a\}$ — множество ошибок, которые необходимо исправить, а $\mathcal{E}^{(2)} = \{\mathbf{E}_a^\dagger \mathbf{E}_b\}$ — множество произведений пар элементов $\mathcal{E}$. Тогда, чтобы построить невырожденный код, который может исправлять ошибки из $\mathcal{E}$, мы должны найти такой набор генераторов стабилизатора, чтобы некоторый генератор антикоммутировал с каждым элементом $\mathcal{E}^{(2)}$.

Чтобы увидеть, может ли выполнить эту работу k-кубитовый код длины n, начнем с множества $\mathcal{S}^{(n-k)}$ всех абелевых подгрупп группы Паули с $n-k$ генераторами. Будем постепенно отбрасывать подгруппы, которые являются неподходящими стабилизаторами для исправления ошибок в $\mathcal{E}$, а затем посмотрим, останется ли что-нибудь.

Каждая нетривиальная ошибка $\mathbf{E}_a$ коммутирует с долей $\sim 1/2^{n-k}$ всех содержащихся в $\mathcal{S}^{(n-k)}$ групп, так как она должна коммутировать с каждым из $n-k$ генераторов группы. (Существует малая поправка к этой доле, которой можно пренебречь при больших n.) Всякий раз, когда мы добавляем к $\mathcal{E}^{(2)}$ еще один элемент, должна быть отброшена доля 2^{k-n} всех кандидатов в стабилизаторы. Когда $\mathcal{E}^{(2)}$ полностью собрана, мы в худшем случае отбросили долю

$$|\mathcal{E}^{(2)}| \cdot 2^{k-n} \tag{7.198}$$

всех содержащихся в $\mathcal{S}^{(n-k)}$ подгрупп (где $|\mathcal{E}^{(2)}|$ — количество элементов в $\mathcal{E}^{(2)}$). До тех пор, пока эта доля меньше единицы, выполняющий эту работу стабилизатор будет существовать при больших n.

Если мы хотим исправить $t = pn$ ошибок, то $\mathcal{E}^{(2)}$ должно содержать операторы с весом, не превышающим $2t$, что позволяет сделать оценку

$$\log_2 |\mathcal{E}^{(2)}| \lesssim \log_2 \left[\binom{n}{2pn} 3^{2pn} \right] \sim n[H_2(2p) + 2p \log_2 3]. \tag{7.199}$$

Следовательно, существуют стабилизирующие коды, исправляющие pn ошибок, с асимптотическим значением $R = k/n$, определяемым неравенством

$$\log_2 |\mathcal{E}^{(2)}| + k - n < 0, \quad \text{или} \quad R < 1 - H_2(2p) - 2p \log_2 3. \tag{7.200}$$

Это (асимптотическая) форма квантовой границы Гилберта–Варшамова.

Отсюда следует, что должны существовать коды с ненулевой скоростью, которые защищают от ошибок, возникающих с любой вероятностью $p < p_{GV} \simeq 0{,}0946$. Для кода, который может защитить от каждого ошибочного оператора с весом $\leqslant pn$, максимальная вероятность ошибки, допускаемая границей Рейнса, равна $1/6$.

Хотя хорошие квантовые коды существуют, явная конструкция семейств хороших кодов представляет собой совсем другую проблему. Действительно, ни одной такой конструкции не известно.

7.15. Некоторые коды, исправляющие многократные ошибки

7.15.1. Каскадные коды

Все ККК0, которые мы явно сконструировали до настоящего момента, имеют $d = 3$ (или $d = 2$) и, следовательно, могут исправить (в лучшем случае) одну ошибку. Теперь мы опишем несколько примеров кодов с большим расстоянием.

Наиболее простым способом построения кодов, которые могут исправить большее количество ошибок, является каскадное соединение кодов, способных исправить одну ошибку. Каскадный код представляет собой код внутри кода. Предположим, что мы имеем два ККК0 с $k = 1$, $[[n_1, 1, d_1]]$-код C_1 и $[[n_2, 1, d_2]]$-код C_2. Представим принадлежащее C_2 кодовое слово длины n_2, построенное в виде когерентной суперпозиции произведений состояний, в которых каждый кубит находится в одном из состояний $|0\rangle$ или $|1\rangle$. Теперь, используя код C_1, заменим каждый кубит закодированным состоянием длины n_1; то есть заменим $|0\rangle$ на $|\bar{0}\rangle$, а $|1\rangle$ на $|\bar{1}\rangle$ кода C_1. Результатом является код с длиной $n = n_1 n_2$, $k = 1$ и с расстоянием не меньшим, чем $d = d_1 d_2$. Будем называть код C_2 «внешним», а C_1 — «внутренним».

Собственно, мы уже обсуждали один пример такой конструкции: 9-кубитовый код Шора. В этом случае внутренним кодом является трехкубитовый код повторения с генераторами стабилизатора

$$\mathbf{ZZ1}, \qquad \mathbf{1ZZ}, \tag{7.201}$$

а внешним — трехкубитовый «фазовый код» с генераторами стабилизатора

$$\mathbf{XX1}, \qquad \mathbf{1XX} \tag{7.202}$$

(код повторения, повернутый преобразованием Адамара). Стабилизатор каскадного кода строится следующим образом. Прежде всего, в него включаются генераторы внутреннего кода. В рассматриваемом примере это пары генераторов, которые действуют на каждый из трех кубитов, содержащихся в данном блоке внешнего кода, то есть $\mathbf{Z}_1\mathbf{Z}_2$, $\mathbf{Z}_2\mathbf{Z}_3$ и так далее — всего шесть генераторов. Затем добавляются генераторы внешнего кода. В рассматриваемом случае это

$$\bar{\mathbf{X}}\bar{\mathbf{X}}\bar{\mathbf{1}}, \qquad \bar{\mathbf{1}}\bar{\mathbf{X}}\bar{\mathbf{X}}, \tag{7.203}$$

где $\bar{\mathbf{1}} = 111$, а $\bar{\mathbf{X}} = \mathbf{XXX}$, то есть пара генераторов повернутого преобразованием Адамара кода повторения (7.202), но с операторами $\mathbf{1}$ и $\mathbf{X}$, замененными на закодированные операторы внутреннего кода. Вы легко распознаете в них восемь генераторов стабилизатора обсуждавшегося ранее кода Шора. В этом случае внутренний и внешний коды имеют единичное расстояние (например, $\mathbf{Z11}$ коммутирует со стабилизатором внутреннего кода), однако каскадный код имеет расстояние $3 > d = d_1 d_2 = 1$. Это случилось потому, что код был так искусно сконструирован, что закодированные операции внутреннего кода с весами 1 и 2 не коммутируют со стабилизатором внешнего кода. (Все было бы иначе, если бы мы состыковали код повторения с самим собой, а не с фазовым кодом!)

Каскадное соединение кода $[[5, 1, 3]]$ с самим собой дает код с расстоянием $d = 9$ (способный исправить четыре ошибки); $n = 25$ является минимальной длиной любого известного кода при $k = 1$ и $d = 9$. (Код $[[n, 1, 9]]$ при $n = 23, 24$ согласовывался бы с границей Рейнса, но неизвестно, существует ли на самом деле такой код).

Стабилизатор каскадного кода $[[25, 1, 9]]$ имеет 24 генератора. Двадцать из них получаются как четыре генератора $\mathbf{M}_{1,2,3,4}$, действующие на каждый из пяти субблоков внешнего кода, а оставшиеся четыре — это закодированные операторы $\bar{\mathbf{M}}_{1,2,3,4}$ внешнего кода. Отметим, что стабилизатор содержит элементы с весом четыре (элементы стабилизатора, действующие на каждый из пяти внутренних кодов); следовательно, код вырожденный. Это типичный пример каскадного кода.

Нет причин ограничиваться двухуровневым каскадированием кодов. Из L ККО с параметрами $[[n_1, 1, d_1]], \ldots, [[n_L, 1, d_L]]$ можно построить иерархический код всего с L уровнями кодов внутри кодов; он имеет длину

$$n = n_1 n_2 \ldots n_L, \tag{7.204}$$

и расстояние

$$d \geqslant d_1 d_2 \ldots d_L, \tag{7.205}$$

В частности, путем L-кратного каскадирования кода $[[5, 1, 3]]$ можно построить код с параметрами

$$[[5^L, 1, 3^L]]. \tag{7.206}$$

Строго говоря, это семейство кодов не может защитить от количества ошибок, пропорционального их длине. Скорее, отношение количества ошибок t, которые могут быть исправлены, к длине n равно

$$\frac{t}{n} \sim \frac{1}{2}\left(\frac{3}{5}\right)^L, \tag{7.207}$$

что стремится к нулю при большом L. Но расстояние d может оказаться обманчивой мерой того, как хорошо работает код. Вполне достаточно, чтобы восстановление, отказывая лишь для некоторых способов выбора $t \ll pn$ ошибок, оставалось успешным для *типичных* способов выбора pn ошибочных кубитов. Действительно, при большом n и $p > 0$ каскадные коды могут исправить pn типичных ошибок.

Фактически, тот способ, которым обычно используются каскадные коды, не в полной мере реализует их возможности исправления ошибок. Чтобы быть конкретнее, рассмотрим код $[[5, 1, 3]]$ в случае, когда каждый из пяти кубитов независимо подвергается действию деполяризующего канала с вероятностью ошибки p (то есть каждая из ошибок $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ возникает с вероятностью $p/3$). Несомненно, восстановление будет успешным, если в блоке возникнет меньше двух ошибок. Следовательно, как в разделе 7.4.2, вероятность сбоя можно ограничить неравенством

$$p_{\text{fail}} \equiv p^{(1)} \leqslant \binom{5}{2} p^2 = 10p^2. \tag{7.208}$$

Теперь рассмотрим работу каскадного кода $[[25, 1, 9]]$. Чтобы облегчить себе жизнь, выполним восстановление простым (но неоптимальным) способом. Сначала произведем восстановление в каждом из пяти субблоков, измеряя $\mathbf{M}_{1,2,3,4}$, чтобы получить синдромы содержащихся в них ошибок. После этого измерим генераторы стабилизатора $\bar{\mathbf{M}}_{1,2,3,4}$ внешнего кода, чтобы получить его синдром и, если он обнаруживает ошибку, к одному из субблоков применим один из закодированных операторов $\bar{\mathbf{X}}$, $\bar{\mathbf{Y}}$ или $\bar{\mathbf{Z}}$.

Для внешнего кода восстановление будет успешным, если повреждено не более одного из субблоков, а вероятность $p^{(1)}$ повреждения субблока ограничена неравенством (7.208); таким образом, для кода $[[25, 1, 9]]$ вероятность отказа процедуры восстановления ограничена сверху

$$p^{(2)} \leqslant 10(p^{(1)})^2 \leqslant 10(10p^2)^2 = 1000p^4. \tag{7.209}$$

Очевидно, что это не самая лучшая процедура, поскольку причиной сбоя могут стать четыре ошибки, если они окажутся по две в двух разных субблоках. Так как код имеет расстояние $d = 9$, существует лучшая процедура, которая всегда будет успешно исправлять четыре ошибки, так что $p^{(2)}$ будет иметь порядок p^5, а не p^4. Тем не менее, эта неоптимальная процедура имеет то преимущество, что она очень легко обобщается (и анализируется) в случае многоуровнего соединения.

Действительно, при наличии L уровней каскадного соединения восстановление начинается на самом глубоком внутреннем уровне и прокла-

дывает путь наружу. Решая рекуррентную систему неравенств

$$p^{(\ell)} \leqslant C\left[p^{(\ell-1)}\right]^2 \tag{7.210}$$

с начальным условием $p^{(0)} = p$, мы приходим к выводу, что

$$p^{(L)} \leqslant \frac{1}{C}(Cp)^{2^L} \tag{7.211}$$

(здесь $C = 10$). Нетрудно видеть, что до тех пор, пока $p < 1/10$, вероятность сбоя можно сделать сколь угодно малой, добавляя к коду достаточное количество уровней.

Мы можем записать

$$p^{(L)} \leqslant p_o \left(\frac{p}{p_o}\right)^{2^L}, \tag{7.212}$$

где $p_o = 1/10$ — оценка *пороговой* (то есть допустимой) вероятности ошибки (ниже мы получим лучшие коды и лучшие оценки этого порога). Отметим: чтобы получить

$$p^{(L)} < \varepsilon \tag{7.213}$$

мы можем выбрать размер блока $n = 5^L$; так что

$$n \leqslant \left\lceil \frac{\log(p_o/\varepsilon)}{\log(p_o/p)} \right\rceil^{\log_2 5}. \tag{7.214}$$

В принципе, каскадный код может дать сбой на высоком уровне при гораздо меньшем, чем $n/10$, количестве ошибок, но они должны быть распределены весьма специальным образом, что совсем не характерно для большого n.

Каскадное кодирование неизвестного квантового состояния может выполняться уровень за уровнем. Например, чтобы закодировать $a|0\rangle + b|1\rangle$ в блоке $[[25, 1, 9]]$, мы можем сначала приготовить состояние $a|\bar{0}\rangle + b|\bar{1}\rangle$ в пятикубитовом блоке, используя описанную ранее схему кодирования, а также приготовить четыре пятикубитовых блока в состоянии $|\bar{0}\rangle$. Состояние $a|\bar{0}\rangle + b|\bar{1}\rangle$ можно закодировать на следующем уровне, снова выполняя эту же схему, но теперь со всеми вентилями, замененными на закодированные, действующие на пятикубитовые блоки. Обсуждение конструкции этих закодированных вентилей можно найти в приложении.

7.15.2. Торические коды

Торические коды представляют собой еще одно семейство кодов, которые, как и каскадные, работают гораздо лучше, чем этого можно было бы ожидать, учитывая значения их расстояний. Их описывает профессор Китаев (который их и открыл).[1]

7.15.3. Коды Рида–Маллера

Другим способом построения кодов, которые могут исправить множество ошибок, является осуществление КШС-конструкции. Вспомним, например, частный случай этой конструкции, в котором используется классический код C, содержащийся в дуальном ему коде (в таком случае говорят, что C является «слабо самодуальным» кодом). В КШС-конструкции с каждым смежным классом C в $C^{\perp}$ ассоциируется кодовое слово. Таким образом, мы получаем квантовый код $[[n, k, d]]$, где n – длина кода C, d – (по крайней мере) расстояние кода $C^{\perp}$, а $k = \dim C^{\perp} - \dim C$. Следовательно, для построения КШС-кодов, исправляющих множество ошибок, необходимы слабо самодуальные классические коды с большим минимальным расстоянием.

Один из классов слабо самодуальных классических кодов представляют собой коды Рида–Маллера. Хотя эти коды не особенно эффективны, они очень удобны, поскольку достаточно легко кодируются. Несложно понять также принцип их работы и математическую структуру.[2]

Чтобы подготовиться к построению кодов Рида–Маллера, рассмотрим булевы функции на m битах

$$f : \{0,1\}^m \to \{0,1\}. \tag{7.215}$$

Существует 2^{2^m} таких функций, образующих множество, которое можно рассматривать как двоичное векторное пространство размерности 2^m. Полезно ввести базис в этом пространстве. Вспомним (см. раздел 6.1), что любая булева функция имеет дизъюнктивную нормальную форму. Так как NOT бита x равно $1 - x$, а OR двух битов x и y можно записать как

$$x \vee y = x + y - xy, \tag{7.216}$$

[1] Детальное описание торических кодов можно найти в статье: A. Kitaev, *Fault-tolerant quantum computation by anyons*, Ann. Phys. (NY) **303**(1) pp. 2–30 (2003), quant-ph/9707021. См. также лекции: A. Kitaev, C. Laumann, *Topological phases and quantum computation*, cond-mat/0904.2771. – *Прим. ред.*

[2] См., например, E. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North Holland Publishing Company, Amsterdam, New York, Oxford (1977), chapter 13; перевод: Ф. Дж. Мак-Вильямс, Н. Дж. А. Слоэн, *Теория кодов, исправляющих ошибки*. – М.: Связь, 1979, гл. 13.

то любая булева функция может быть представлена в виде полинома от m двоичных переменных $x_{m-1}, x_{m-2}, \ldots, x_1, x_0$. Базис векторного пространства полиномов состоит из 2^m функций

$$1, \quad x_i, \quad x_i x_j, \quad x_i x_j x_k, \ldots, \tag{7.217}$$

(где каждый моном представляет собой произведение различных сомножителей, так как $x^2 = x$). Каждую такую функцию f можно представить двоичной строкой длины 2^m, значение которой в позиции, обозначенной двоичной строкой $x_{m-1}, x_{m-2}, \ldots, x_1, x_0$, равно $f(x_{m-1}, x_{m-2}, \ldots, x_1, x_0)$. Например, для $m = 3$

$$\begin{aligned} 1 &= (11111111), \\ x_0 &= (10101010), \\ x_1 &= (11001100), \\ x_2 &= (11110000), \\ x_0 x_1 &= (10001000), \\ x_0 x_2 &= (10100000), \\ x_1 x_2 &= (11000000), \\ x_0 x_1 x_2 &= (10000000). \end{aligned} \tag{7.218}$$

Подпространство этого векторного пространства получается, если ограничить степень полинома до r или менее. Это подпространство является кодом Рида–Маллера (или РМ-кодом) и обозначается $R(r, m)$. Его длина равна $n = 2^m$, а его размерность

$$k = 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}. \tag{7.219}$$

Некоторые частные случаи, представляющие интерес:

- $R(0, m)$ — код повторения длины 2^m.
- $R(m-1, m)$ — код, дуальный коду повторения, пространство всех строк четного веса длины 2^m.
- $R(1, 3)$ — код, натянутый на $1, x_0, x_1, x_2$ с параметрами $n = 8$, $k = 4$; фактически, это уже обсуждавшийся расширенный код Хэмминга $[8, 4, 4]$.
- В более общем случае $R(m-2, m)$ при любом $m \geqslant 3$ представляет собой расширенный код Хэмминга с $d = 4$. Выкалывая его (убирая последний бит из каждого кодового слова), мы получим совершенный код Хэмминга $[n = 2^m - 1, k = n - m, d = 3]$.

- $R(1,m)$ имеет $d = 2^{m-1} = \frac{1}{2}n$ и $k = m$. Он дуален расширенному коду Хэмминга и известен как «код Рида–Маллера первого порядка». Он сам по себе представляет значительный практический интерес, благодаря его большому расстоянию и особой простоте процедуры декодирования.

Применяя метод индукции по m, можно вычислить расстояние кода $R(r,m)$. Сначала мы должны определить, как $R(r,m+1)$ связан с $R(r,m)$. Функцию от $x_m,\ldots,x_0$ можно записать как

$$f(x_m,\ldots,x_0) = g(x_{m-1},\ldots,x_0) + x_m h(x_{m-1},\ldots,x_0), \tag{7.220}$$

если f имеет степень r, тогда g должна иметь степень r, а h — степень $r-1$. Рассматривая f как вектор длины 2^{m+1}, имеем[1]

$$f = (g|g) + (h|0), \tag{7.221}$$

где g,h — векторы длиной 2^m. Рассмотрим расстояние между f и

$$f' = (g'|g') + (h'|0). \tag{7.222}$$

При $h = h'$ и $f \neq f'$ это расстояние равно $\mathrm{wt}(f-f') = 2\cdot\mathrm{wt}(g-g') \geqslant 2\cdot\mathrm{dist}(R(r,m))$; при $h \neq h'$ оно по крайней мере $\mathrm{wt}(h-h')) \geqslant \mathrm{dist}(R(r-1,m))$. Если $d(r,m)$ обозначает расстояние кода $R(r,m)$, то можно видеть, что

$$d(r,m+1) = \min[2d(r,m); d(r-1,m)]. \tag{7.223}$$

Теперь с помощью индукции по m можно показать, что $d(r,m) = 2^{m-r}$. Во-первых, проверим, что $d(r,m=1) = 2^{1-r}$ при $r = 0,1$; $R(1,1)$ представляет собой пространство всех строк длины два, а $R(0,1)$ — код повторения длины два. Предположим теперь, что при всех $m \leqslant M$ и $0 \leqslant r \leqslant m$ расстояние $d = 2^{m-r}$. Тогда мы делаем вывод, что при всех $1 \leqslant r \leqslant m$

$$d(r,m+1) = \min(2^{m-r+1}; 2^{m-r+1}) = 2^{m-r+1}. \tag{7.224}$$

[1]Здесь символ $(f|g)$ обозначает следующую конструкцию: если $f = (f_1, f_2 \cdots, f_n)$ — вектор-строка длины n, а $g = (g_1, g_2 \cdots, g_m)$ — вектор-строка длины m, то $(f|g) = (f_1, f_2 \cdots, f_n, g_1, g_2 \cdots, g_m)$ — вектор-строка длины $n+m$. Вывод выражения (7.221) для вектора-строки длины 2^{m+1}, изображающей булеву функцию (7.220), а также подробности вычисления расстояния $d(r,m)$ кода Рида–Маллера $R(r,m)$ можно найти в книге: E. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North Holland Publishing Company, Amsterdam, New York, Oxford (1977), chapter 13; перевод: Ф. Дж. Мак-Вильямс, Н. Дж. А. Слоэн, *Теория кодов, исправляющих ошибки*. — М.: Связь, 1979, гл. 13. — *Прим. ред.*

Также ясно, что $d(m+1, m+1) = 1$, так как $R(m+1, m+1)$ представляет собой пространство всех двоичных строк длины 2^{m+1}, и что $d(0, m+1) = 2^{m+1}$, так как $R(0, m+1)$ является кодом повторения длины 2^{m+1}. Это завершает индуктивный шаг и доказывает, что $d(r, m) = 2^{m-r}$.

Отсюда, в частности, следует, что $R(m-1, m)$ имеет расстояние два, поэтому дуальным коду $R(r, m)$ является $R(m-r-1, m)$. Прежде всего, заметим, что сумма биномиальных коэффициентов $\binom{m}{j}$ $(0 \leqslant j \leqslant m)$ равна 2^m, так что $R(m-r-1, m)$ имеет правильную размерность, чтобы иметь смысл $R^{\perp}(r, m)$. Тогда этого достаточно, чтобы показать, что $R(m-r-1, m)$ содержится в $R(r, m)$. Но если $f \in R(r, m)$, а $g \in R(m-r-1, m)$, то их произведение является полиномом степени не выше $m-1$ и, следовательно, принадлежит $R(m-1, m)$. Каждый вектор в $R(m-1, m)$ имеет четный вес, поэтому внутреннее произведение $f \cdot g$ обращается в нуль; следовательно, g принадлежит дуальному пространству $R^{\perp}(r, m)$. Это показывает, что

$$R^{\perp}(r, m) = R(m-r-1, m). \tag{7.225}$$

Именно благодаря этому замечательному свойству дуальности коды Рида–Маллера хорошо подходят для КШС-конструкции квантовых кодов.

В частности, код Рида–Маллера является слабо самодуальным при $r \leqslant m-r-1$, или $2r \leqslant m-1$, и самодуальным при $2r = m-1$. В последнем случае его расстояние равно

$$d = 2^{m-r} = 2^{\frac{1}{2}(m+1)} = \sqrt{2n}, \tag{7.226}$$

а количество закодированных битов —

$$k = \frac{1}{2}n = 2^{m-1}. \tag{7.227}$$

При $m = 3, 5, 7$ эти самодуальные коды имеют параметры

$$[8, 4, 4], \quad [32, 16, 8], \quad [128, 64, 16]. \tag{7.228}$$

(Как уже отмечалось, код [8,4,4] является расширенным кодом Хэмминга.) С ними связаны квантовые коды с параметрами $(k = 0)$

$$[[8, 0, 4]], \quad [[32, 0, 8]], \quad [[128, 0, 16]] \tag{7.229}$$

и так далее.

Для того чтобы получить квантовый код с $k = 1$, достаточно *выколоть* самодуальный код Рида–Маллера, то есть удалить из него один из $n = 2^m$

битов (какой бит удалить, значения не имеет). Результатом этой операции является классический код с параметрами $n = 2^m - 1$, $d = 2^{(m+1)/2} - 1 = \sqrt{2(n+1)} - 1$ и $k = (n+1)/2$. Более того, дуальным этому выколотому коду является его четный субкод. (Четный субкод состоит из тех РМ-кодовых слов, для которых удаляемый при выкалывании бит равен нулю, а из самодуальности РМ-кода следует, что они ортогональны всем словам (с четным и нечетным весами) выколотого кода.) Из этих выколотых кодов с помощью КШС-конструкции мы получаем квантовые коды с параметрами $(k = 1)$

$$[[7, 1, 3]], \quad [[31, 1, 7]], \quad [[127, 1, 15]] \tag{7.230}$$

и так далее. Код Хэмминга $[7, 4, 3]$ получается при выкалывании РМ-кода $[8, 4, 4]$, а ККК0, соответствующий коду $[7, 1, 3]$, естественно, является кодом Стина. Эти ККК0 имеют расстояние, возрастающее как квадратный корень их длины.

Эти коды с $k = 1$ не самые эффективные из известных ККК0. Тем не менее, они представляют особый интерес, так как их свойства особенно подходят для применения помехоустойчивых квантовых вентилей на закодированную информацию (см. приложение). В частности, одним полезным свойством самодуальных РМ-кодов является их «двойная четность» — все кодовые слова имеют кратный четырем вес.

Конечно, применяя КШС-конструкцию к РМ-кодам, мы также можем построить квантовые коды с $k > 1$. Например, $R(3, 6)$ с параметрами

$$\begin{aligned} n &= 2^m = 64, \\ d &= 2^{m-r} = 8, \\ k &= 1 + \binom{6}{1} + \binom{6}{2} + \binom{6}{3} = 1 + 6 + 15 + 20 = 42 \end{aligned} \tag{7.231}$$

дуален по отношению к $R(2, 6)$ с параметрами

$$\begin{aligned} n &= 2^m = 64, \\ d &= 2^{m-r} = 16, \\ k &= 1 + \binom{6}{1} + \binom{6}{2} = 1 + 6 + 15 = 22, \end{aligned} \tag{7.232}$$

и, следовательно, КШС-конструкция дает ККК0 с параметрами

$$[[64, 20, 8]]. \tag{7.233}$$

Известны многие другие слабо самодуальные коды, которые можно использовать таким же образом.

7.15.4. Код Голея

С точки зрения чистой математики, самым интересным из когда-либо открытых корректирующих ошибки кодов (классических или квантовых) является код Голея, который был еще и одним из первых, описанных в открытой печати. Здесь мы кратко опишем его, поскольку с помощью КШС-конструкции этот код тоже может трансформироваться в хороший ККК0. (Возможно, этот ККК0 на самом деле не настолько важен, чтобы посвящать ему раздел в этой главе; и все же он достаточно занятный, так что я включил его сюда.)

Код Голея (расширенный) представляет собой самодуальный классический код $[24, 12, 8]$. Если мы выколем его (удалим любой из его 24-х битов), то получим код Голея $[23, 12, 7]$, который может исправить три ошибки. Этот код на самом деле является совершенным, так как он насыщает границу упаковки сфер:

$$1 + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = 2^{11} = 2^{23-12}. \tag{7.234}$$

На самом деле, совершенные коды, исправляющие больше одной ошибки, — невероятная редкость. Можно показать,[1] что способными исправить больше одной ошибки совершенными кодами (линейными или нелинейными) над *любым* конечным полем являются *всего лишь два*: код $[23, 12, 7]$ и еще один открытый Голеем двоичный код с параметрами $[11, 6, 5]$.

Код Голея $[24, 12, 8]$ имеет очень сложную симметрию. Она характеризуется своей группой автоморфизмов — группой перестановок 24-х битов, преобразующих одни кодовые слова в другие. Это группа Матье M_{24}, открытая в XIX веке спорадическая простая группа порядка 244 823 040.

$2^{12} = 4096$ кодовых слов имеют распределение весов (в очевидном обозначении)

$$0^1 8^{759} 12^{2576} 16^{759} 24^1. \tag{7.235}$$

Отметим, в частности, что каждый вес кратен четырем (код имеет двойную четность). Каков смысл числа $759 (= 3 \cdot 11 \cdot 23)$? На самом деле оно равно

$$\binom{24}{5} \Big/ \binom{8}{5} = 759 \tag{7.236}$$

[1] См. § 6.10 в книге E. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North Holland Publishing Company, Amsterdam, New York, Oxford (1977); перевод: Ф. Дж. МакВильямс, Н. Дж. Слоэн, *Теория кодов, исправляющих ошибки*. — М.: Связь, 1979.

и возникает по комбинаторной причине: каждое кодовое слово с весом восемь характеризуется своим носителем — 8-элементным множеством («октадой»). Последние выбираются таким образом, чтобы каждое 5-элементное подмножество 24-х битов содержалось (целиком) в одной и только одной такой октаде (отражение высокой симметрии кода).

Что придает коду Голея математическую значимость? Его открытие в 1949 году привело в движение последовательность событий, которые примерно к 1980 году завершились полной классификацией конечных простых групп. Эта классификация является одним из величайших достижений математики XX века.

(Группа является простой, если она не содержит ни одной нетривиальной нормальной подгруппы. Конечные простые группы можно рассматривать как строительные блоки всех конечных групп в том смысле, что для любой конечной группы G существует однозначное разложение вида

$$G \equiv G_0 \supseteq G_1 \supseteq G_2 \supseteq \ldots \supseteq G_n, \tag{7.237}$$

где каждая G_{j+1} представляет собой нормальную подгруппу G_j, а каждая фактор-группа G_j/G_{j+1} является простой. Конечные простые группы можно систематизировать в разные бесконечные семейства, плюс 26 дополнительных не поддающихся классификации «спорадических» простых групп.)

В 1964 году код Голея привел Лича к открытию чрезвычайно плотной укладки шаров в 24-х измерениях, известной как *решетка Лича* Λ. Узлы решетки (центры сфер) представляют собой 24-компонентные целочисленные векторы со следующими свойствами: чтобы определить, содержится ли $\vec{x} = (x_1, x_2, \ldots x_{24})$ в Λ, запишем каждую компоненту x_j в двоичном представлении

$$x_j = \ldots x_{j^3} x_{j^2} x_{j^1} x_{j^0}. \tag{7.238}$$

Тогда $\vec{x} \in \Lambda$, если[1]

(i) все x_{j^0} либо нули, либо единицы;

(ii) x_{j^2} представляют собой четную 24-битовую строку, если все x_{j^0} равны нулю, и — нечетную 24-битовую строку, если все x_{j^0} равны единице;

[1]Некоторые альтернативные определения решетки Лича можно найти в книге J. H. Conway, N. J. A. Sloane, *Sphere Packing, Lattices and Groups*, Springer Verlag, NY, Berlin, et al. (1988), Chapter 4, § 11; перевод: Дж. Конвей, Н. Слоэн, *Упаковки шаров, решетки и группы*. — М.: Мир, 1990, глава 4, § 11.— *Прим. ред.*

(iii) x_{j^1} представляют собой 24-битовую строку, содержащуюся в коде Голея.

При употреблении этих правил отрицательное число представляется его двоичным дополнением, например,

$$\begin{aligned} -1 &= \ldots 1111, \\ -2 &= \ldots 1110, \\ -3 &= \ldots 1101, \\ &\text{и так далее.} \end{aligned} \tag{7.239}$$

Нетрудно проверить, что Λ является решеткой: она замкнута относительно сложения. (На остальные биты, кроме битов последних трех разрядов двоичного разложения x_j, никаких ограничений не накладывается.)

Подсчитаем число ближайших к началу координат[1] соседей (или количество сфер, касающихся любой данной сферы). Все эти точки находятся на расстоянии $(\text{distance})^2 = 32$ от начала координат

$$\begin{aligned} (\pm 2)^8(0)^{16} &: 2^7 \cdot 759, \\ (\pm 3)(\mp 1)^{23} &: 2^{12} \cdot 24, \\ (\pm 4)^2(0)^{22} &: 2^2 \cdot \binom{24}{2}. \end{aligned} \tag{7.240}$$

Таким образом существует $2^7 \cdot 759$ ближайших соседей, радиус-векторы которых имеют восемь отличных от нуля компонент, равных ± 2 (среди них количество отрицательных — четное) и заполняющих позиции, принадлежащие носителям 759 кодовых слов Голея с весом восемь. Далее, существует $2^{12} \cdot 24$ ближайших соседей, радиус-векторы которых имеют одну компоненту, равную ± 3 (она может находиться в любой из 24-х позиций), а остальные 23 компоненты равны ∓ 1. При этом верхние знаки приписываются компонентам, которые заполняют позиции, принадлежащие носителям кодовых слов Голея произвольного веса. Если, например, выбрана компонента $+3$, то занимаемая ею позиция вместе с позициями всех отрицательных компонент (то есть -1) образует носитель одного из 2^{11} оставшихся (из 2^{12}) кодовых слов Голея. Наконец, имеется $2^2 \cdot \binom{24}{2}$ ближайших соседей, радиус-векторы которых имеют только две отличные от нуля компоненты ± 4, положение и знак которых ничем не ограничены. В сумме координационное число решетки равно 196 560.

[1] Символ $(a)^k(b)^l$ обозначает вектор решетки $\vec{x} \in \Lambda$, имеющий k и l компонент, равных a и, соответственно, b. — *Прим. ред.*

Решетка Лича имеет замечательную группу автоморфизмов, открытую Конвэем в 1968 году. Это сохраняющая решетку конечная подгруппа группы вращений $\mathrm{SO}(24)$ пространства размерности 24. Порядок этой конечной группы (известной как $\cdot 0$, или «точка нуль») равен

$$2^{22} \cdot 3^9 \cdot 5^4 \cdot 7^2 \cdot 11 \cdot 13 \cdot 23 = 8\,315\,553\,613\,086\,720\,000 \simeq 8.3 \times 10^{18}. \quad (7.241)$$

Если ее двухэлементный центр выкидывается, получается спорадическая простая группа $\cdot 1$. К моменту ее открытия, $\cdot 1$ была самой большой из построенных спорадических простых групп.

Решетка Лича и ее группа автоморфизмов, в конечном счете, (путем, который не будет здесь описан) в 1982 году привели Грисса к построению наиболее удивительной из всех спорадической простой группы (на существование которой ранее указывали Фишер и Грисс). Это конечная подгруппа группы вращений в пространстве размерности 196 884, порядок которой приблизительно равен 8.08×10^{53}. Это чудище, известное как F_1, получило прозвище «монстр» (хотя Грисс предпочитает называть его «дружелюбным гигантом»). Открытая последней, она является самой большой спорадической простой группой.

Таким образом, классификация конечных простых групп многим обязана (классической) теории кодирования и, в частности, коду Голея. Возможно, теория ККК0 также завещает математике что-нибудь значительное и очень интересное!

Во всяком случае, поскольку (расширенный) код Голея $[24, 12, 8]$ является самодуальным, то получаемый при выкалывании код $[23, 12, 7]$ — слабо самодуальный; дуальный ему код $[23, 11, 8]$ является его собственным субкодом. Отсюда с помощью метода КШС можно построить ККК0 $[[23, 1, 7]]$. Это не самый эффективный квантовый код, который может исправить три ошибки (существует код $[[17, 1, 7]]$, насыщающий границу Рейнса), но он обладает особенно тонкими свойствами, благоприятствующими помехоустойчивым квантовым вычислениям (см. приложение).

7.16. Пропускная способность квантового канала

Как это до сих пор формулировалось, целью построения ККК0 является достижение максимального значения расстояния кода d, при заданных длине n и количестве закодированных кубитов k. Большее расстояние обеспечивает лучшую защиту от ошибок, так как код с расстоянием d может исправить $d - 1$ стираний или $(d - 1)/2$ ошибок в неизвестных позициях. Мы видели, что можно построить хорошие коды, поддерживающие конеч-

ную скорость воспроизведения k/n при большом n и корректирующие pn ошибок, количество которых пропорционально n.

Теперь мы обратимся к другому, но достаточно близкому вопросу об асимптотическом исполнении ККО. Пусть \$ — супероператор, действующий на операторы плотности в гильбертовом пространстве $\mathcal{H}$. Будем рассматривать супероператоры \$, действующие независимо в каждой копии $\mathcal{H}$, содержащейся в n-кратном тензорном произведении

$$\mathcal{H}^{(n)} = \mathcal{H} \otimes \ldots \otimes \mathcal{H}. \tag{7.242}$$

Мы бы хотели выбрать такое кодовое подпространство $\mathcal{H}_{\text{code}}^{(n)}$ пространства $\mathcal{H}^{(n)}$, чтобы содержащаяся в $\mathcal{H}_{\text{code}}^{(n)}$ квантовая информация подвергалась действию супероператора

$$\$^{(n)} = \$ \otimes \ldots \otimes \$ \tag{7.243}$$

и, тем не менее, могла быть декодирована с высокой точностью воспроизведения.

Скорость воспроизведения кода определяется как

$$R = \frac{\log \mathcal{H}_{\text{code}}^{(n)}}{\log \mathcal{H}^{(n)}}; \tag{7.244}$$

это количество кубитов, предназначенных для переноса одного кубита закодированной информации. *Пропускная способность квантового канала* $Q(\$)$ супероператора \$ представляет собой максимум асимптотической скорости воспроизведения, при которой квантовую информацию можно послать по каналу со сколь угодно высокой точностью воспроизведения. Другими словами, $Q(\$)$ является таким наибольшим числом, что для любого $R < Q(\$)$ и любого $\varepsilon > 0$ существует такой код $\mathcal{H}_{\text{code}}^{(n)}$ со скоростью воспроизведения, по крайней мере равной R, что для любого $|\psi\rangle \in \mathcal{H}_{\text{code}}^{(n)}$ состояние $\boldsymbol{\rho}$, *восстановленное после того, как* $|\psi\rangle$ подвергалось действию $\$^{(n)}$, имеет точность воспроизведения

$$F = \langle\psi|\boldsymbol{\rho}|\psi\rangle > 1 - \varepsilon. \tag{7.245}$$

Таким образом, $Q(\$)$ представляет собой квантовую версию определенной Шенноном пропускной способности классического канала с шумом. Как мы уже видели в пятой главе, это не единственный вид пропускной способности, которую можно связать с квантовым каналом. Также большой интерес представляет $C(\$)$ — максимальная скорость, с которой *классическую* информацию можно передавать по квантовому каналу

со сколь угодно малой вероятностью ошибки. Формальный ответ на этот вопрос был сформулирован в разделе 5.4, но только для ограниченного класса возможных схем кодирования; общий ответ до сих пор неизвестен. Пропускная способность квантового канала $Q(\$)$ даже еще менее понятна, чем классическая пропускная способность $C(\$)$ квантового канала. Отметим, что $Q(\$)$ и максимальная асимптотическая скорость воспроизведения k/n, которая может быть достигнута хорошими $[[n, k, d]]$ ККК0 с положительным d/n, суть не одно и то же. В случае пропускной способности квантового канала мы не должны требовать, чтобы код корректировал *любое* возможное распределение pn ошибок, при условии, что ошибки, которые невозможно исправить, становятся в высшей степени атипичными при большом n.

Здесь мы в основном ограничимся обсуждением двух интересных примеров квантовых каналов, действующих на одиночный кубит — квантового стирающего канала (для которого точное значение Q известно) и деполяризующего канала (для которого Q не известна до сих пор, но для нее можно установить полезные верхнюю и нижнюю границы).

Что это за каналы? В случае квантового стирающего канала, переданный кубит либо приходит неповрежденным, либо (с вероятностью p) теряется и его никогда не получают. Мы можем найти унитарное представление этого канала, погружая кубит в трехмерное гильбертово пространство с ортонормированным базисом $\{|0\rangle, |1\rangle, |2\rangle\}$. Канал действует согласно правилу

$$\begin{aligned} |0\rangle \otimes |0\rangle_E &\rightarrow \sqrt{1-p}|0\rangle \otimes |0\rangle_E + \sqrt{p}|2\rangle \otimes |1\rangle_E, \\ |1\rangle \otimes |0\rangle_E &\rightarrow \sqrt{1-p}|1\rangle \otimes |0\rangle_E + \sqrt{p}|2\rangle \otimes |2\rangle_E, \end{aligned} \tag{7.246}$$

где $\{|0\rangle_E, |1\rangle_E, |2\rangle_E\}$ — взаимно ортогональные состояния окружения. Получатель может измерить наблюдаемую $|2\rangle\langle 2|$, чтобы определить, остался ли кубит неповрежденным или был «стерт».

Деполяризующий канал (с вероятностью ошибки p) детально обсуждался в разделе 3.4.1. Мы видим, что при $p \leqslant 3/4$ судьбу переданного по каналу кубита можно описать следующим образом: с вероятностью $1 - q$ (где $q = 4p/3$) кубит доходит неповрежденным, а с вероятностью q — разрушается; в последнем случае его состояние описывается случайной матрицей плотности $\frac{1}{2}1$.

И стирающий, и деполяризующий каналы разрушают кубит с определенной вероятностью. Их главное различие состоит в том, что в случае стирающего канала получатель знает, какие кубиты были разрушены; в случае деполяризующего канала поврежденные кубиты не несут никаких способствующих восстановлению отличительных признаков. Конечно, в обоих

случаях, отправитель не может заранее знать, какие кубиты будут уничтожены.

7.16.1. Стирающий канал

Пропускную способность квантового стирающего канала можно точно определить. Сначала мы установим верхнюю границу для Q, а затем покажем, что существуют коды, достигающие высокой точности воспроизведения и сколь угодно близкой к верхней границе скорости воспроизведения. На первом этапе вывода верхней границы пропускной способности покажем, что $Q = 0$ при $p > 1/2$.

Стирающий канал может быть реализован, если Алиса посылает кубит Бобу, а третья сторона в лице Чарли решает случайным образом, украсть кубит (с вероятностью p) или позволить кубиту дойти до Боба неповрежденным (с вероятностью $1-p$). Если Алиса посылает большое количество кубитов n, то примерно $(1-p)n$ кубитов доходят до Боба, а pn — перехватываются Чарли. Следовательно, при $p > 1/2$ Чарли обладает бóльшим количеством кубитов, чем Боб, и если Боб может восстановить закодированную Алисой квантовую информацию, то вне всякого сомнения это может сделать и Чарли. Следовательно, если $Q(p) > 0$ при $p > 1/2$, Боб и Чарли могут клонировать отправленные Алисой неизвестные закодированные квантовые состояния, что невозможно. (Строго говоря, они могут клонировать с точностью воспроизведения $F = 1 - \varepsilon$, для любого $\varepsilon > 0$.) Таким образом, при $p > 1/2$ пропускная способность квантового канала $Q(p) = 0$.

Чтобы найти границу для $Q(p)$ в случае $p < 1/2$, мы обратимся к следующей лемме. Предположим, Алиса и Боб связаны посредством идеального канала и канала с шумом с пропускной способностью $Q > 0$. Допустим также, что Алиса посылает m кубитов по идеальному каналу и n кубитов по каналу с шумом. Тогда количество закодированных кубитов r, которые Боб может восстановить со сколь угодно высокой точностью воспроизведения, должно удовлетворять неравенству

$$r \leqslant m + Qn. \tag{7.247}$$

Мы получим его, заметив, что Алиса и Боб могут имитировать m отправленных по идеальному каналу кубитов, посылая m/Q кубитов по каналу с шумом и таким образом достигая скорости воспроизведения

$$R = \frac{r}{m/Q + n} = \left(\frac{r}{m + Qn}\right) Q. \tag{7.248}$$

Если бы r превысило $m + Qn$, эта скорость R превысила бы пропускную способность канала с шумом Q, что невозможно. Следовательно, справедливо неравенство (7.247).

Теперь рассмотрим стирающий канал с вероятностью ошибки p_1 и предположим, что $Q(p_1) > 0$. Тогда для $p_2 \leqslant p_1$ мы можем ограничить $Q(p_2)$ неравенством

$$Q(p_2) \leqslant 1 - \frac{p_2}{p_1} + \frac{p_2}{p_1} Q(p_1). \tag{7.249}$$

(Другими словами, если мы строим график $Q(p)$ на плоскости (p, Q) и проводим секущую линию из любой точки (p_1, Q_1) до точки $(p = 0, Q = 1)$, то в интервале $0 \leqslant p \leqslant p_1$ кривая $Q(p)$ не может лежать выше секущей; если $Q(p)$ дважды дифференцируема, то ее вторая производная не может быть отрицательной.) Чтобы получить эту границу, представим, что Алиса посылает Бобу n кубитов, заранее зная, какие $n(1-p_2/p_1)$ из них прибудут неповрежденными. Оставшиеся $n(p_2/p_1)$ кубитов стираются с вероятностью p_1. Следовательно, Алиса и Боб используют как идеальный канал, так и канал с шумом с вероятностью стирания p_1; неравенство (7.247) верно, а скорость воспроизведения R, которой они могут достичь, ограничена неравенством

$$R \leqslant 1 - \frac{p_2}{p_1} + \frac{p_2}{p_1} Q(p_1). \tag{7.250}$$

С другой стороны, при больших n стирается всего около np_2 кубитов, а $(1-p_2)n$ кубитов доходят неповрежденными. Таким образом, Алиса и Боб имеют стирающий канал с вероятностью стирания p_2, но с тем дополнительным преимуществом, что они заранее знают, что некоторые из отправленных Алисой кубитов неуязвимы для стирания. Располагая этой информацией, они оказываются в менее затруднительном положении, чем без нее; отсюда следует (7.249). Эта же граница применима и для деполяризующего канала.

Теперь результат $Q(p) = 0$ при $p > 1/2$ можно скомбинировать с неравенством (7.249). Мы делаем вывод, что кривая $Q(p)$ не может находиться выше прямой линии, соединяющей точки $(p=0, Q=1)$ и $(p=1/2, Q=0)$, или

$$Q(p) \leqslant 1 - 2p, \qquad 0 \leqslant p \leqslant \frac{1}{2}. \tag{7.251}$$

Действительно, существуют стабилизирующие коды, которые фактически достигают скорости воспроизведения $1 - 2p$ при $0 \leqslant p \leqslant 1/2$. Это можно увидеть, позаимствовав идею Клода Шеннона и усреднив по случайным стабилизирующим кодам. Представим выбор (подряд) всех $n - k$

генераторов стабилизатора. Каждый выбирается среди 4^n операторов Паули, имеющих одинаковую априорную вероятность, за исключением того, что каждый новый генератор должен коммутировать со всеми выбранными в предыдущих раундах генераторами.

Теперь Алиса использует этот стабилизирующий код, чтобы закодировать произвольное квантовое состояние в 2^k-мерном кодовом подпространстве, и посылает Бобу n кубитов по стирающему каналу с вероятностью стирания p. Сможет ли Боб восстановить отправленное Алисой состояние?

Боб заменяет каждый стертый кубит кубитом в состоянии $|0\rangle$, а затем приступает к измерению всех $n-k$ генераторов стабилизатора. Из этого измерения синдромов он надеется извлечь оператор Паули $\mathbf{E}$, действующий на замещенные кубиты. Коль скоро $\mathbf{E}$ известен, он может применить $\mathbf{E}^\dagger$, чтобы восстановить идеальную копию отправленного Алисой состояния. При большом n количество кубитов, которые Боб должен заменить, примерно равно pn, и он успешно их восстановит, если существует единственный оператор Паули $\mathbf{E}$, производящий искомый синдром. Если один и тот же синдром имеет более одного оператора Паули, действующего на замещенные кубиты, то восстановление может не удаться.

Какова вероятность сбоя? Так как мы имеем около pn замещенных кубитов, существует около 4^{pn} операторов Паули с носителем на этих кубитах. Более того, для любого конкретного оператора Паули $\mathbf{E}$ случайный стабилизирующий код генерирует случайный синдром — каждый генератор *стабилизатора с вероятность* $1/2$ коммутирует с $\mathbf{E}$ и с такой же вероятностью антикоммутирует. Следовательно, вероятность того, что два оператора Паули имеют одинаковый синдром, равна $(1/2)^{n-k}$.

Существует по крайней мере один действующий на замещенные кубиты особый оператор Паули, который имеет искомый Бобом синдром. Но вероятность того, что другой оператор Паули имеет такой же синдром (а следовательно, вероятность сбоя), не более, чем

$$P_{\text{fail}} \leqslant 4^{pn}\left(\frac{1}{2}\right)^{n-k} = 2^{-n(1-2p-R)}. \tag{7.252}$$

где $R = k/n$ — скорость воспроизведения. Неравенство (7.252) ограничивает вероятность сбоя, если мы *усредняем* по всем стабилизирующим кодам со скоростью R; отсюда следует, что должен существовать по крайней мере один стабилизирующий код, вероятность сбоя которого также удовлетворяет этому неравенству.

Для этого конкретного кода P_{fail} становится сколь угодно малой при $n \to \infty$, при любой скорости воспроизведения $R = 1 - 2p - \delta$, строго

меньшей $1-2p$. Следовательно, $R=1-2p$ асимптотически достижима; объединяя этот результат с неравенством (7.251), мы получаем пропускную способность квантового стирающего канала

$$Q(p)=1-2p, \qquad 0\leqslant p\leqslant \frac{1}{2}. \tag{7.253}$$

Если бы мы хотели гарантировать, что каждому способу повреждения pn стертых кубитов можно сопоставить определенный синдром, тогда нам понадобился бы квантовый код $[[n,k,d]]$ с расстоянием $d>pn$. Граница Гилберта–Варшамова из раздела 7.14 гарантирует существование такого кода при

$$R<1-H_2(p)-p\log_2 3. \tag{7.254}$$

Эта скорость может быть достигнута кодом, который защищает от всех возможных способов стирания до pn кубитов. При $p>0$ она лежит строго ниже пропускной способности, потому что для достижения высокой средней точности воспроизведения достаточно быть способным исправлять *типичные* стирания, а не все возможные ошибки.

7.16.2. Деполяризующий канал

Пропускная способность канала деполяризации до сих пор точно не известна, но мы можем получить для нее некоторые интересные верхнюю и нижнюю границы.

Как и в случае стирающего канала, мы можем найти верхнюю границу для пропускной способности, прибегая к теореме о невозможности клонирования. Вспомним, что для деполяризующего канала с вероятностью ошибки $p<3/4$ каждый кубит с вероятностью $1-4p/3$ проходит неповрежденным, либо с вероятностью $q=4p/3$ рандомизируется (заменяется максимально смешанным $\boldsymbol{\rho}=\frac{1}{2}\mathbf{1}$). Тогда подслушивающий Чарли может имитировать канал, с вероятностью q перехватывая кубиты и замещая каждый украденный кубит максимально смешанным кубитом. При $q>1/2$ Чарли перехватывает больше половины кубитов и находится в более выгодном, чем Боб, положении для декодирования отправленного Алисой состояния. Следовательно, чтобы не позволить клонирование, скорость, с которой Алиса посылает Бобу квантовую информацию, должна быть строго нулевой при $q>1/2$ или $p>3/8$:

$$Q(p)=0, \qquad p>\frac{3}{8}. \tag{7.255}$$

На самом деле можно получить более строгую границу, заметив, что Чарли может избрать лучшую стратегию подслушивания — применить оптимальный *приближенный* клонер, который вы изучили в домашней задаче. Это устройство применяется к каждому отправленному Алисой кубиту и заменяет его двумя кубитами, так что каждый приближается к оригиналу с точностью воспроизведения $F = 5/6$, или

$$|\psi\rangle\langle\psi| \to \left[(1-q)|\psi\rangle\langle\psi| + q\frac{1}{2}\mathbf{1}\right]^{\otimes 2}, \tag{7.256}$$

где $F = 5/6 = 1 - q/2$. Управляя клонером, Чарли и Боб могут получить состояние Алисы, переданное по деполяризующему каналу с $q = 1/3$. Следовательно, достижимая скорость воспроизведения должна стремиться к нулю; иначе, объединяя приближенный клонер и квантовую коррекцию ошибок, Боб и Чарли смогли бы точно клонировать неизвестное состояние Алисы. Таким образом, уже при $q > 1/3$ или $p > 1/4$ пропускная способность должна обращаться в нуль:

$$Q(p) = 0, \qquad p > \frac{1}{4}. \tag{7.257}$$

Учитывая границу (7.249), мы приходим к выводу, что

$$Q(p) \leqslant 1 - 4p, \qquad 0 \leqslant p \leqslant \frac{1}{4}. \tag{7.258}$$

Этот результат фактически совпадает с найденной в разделе 7.8 границей для скорости воспроизведения кодов $[[n, k, d]]$ при $k \geqslant 1$ и $d \geqslant 2pn + 1$. Предел для пропускной способности и граница для допустимой вероятности ошибки кода $[[n, k, d]]$ (а в последнем случае граница Рейнса является более строгой) — это разные понятия. Тем не менее, сходство между ними не так уж и удивительно, поскольку обе эти границы выводятся из теоремы о невозможности клонирования.

Мы можем получить нижнюю границу для пропускной способности, приблизительно подсчитав скорость воспроизведения, которая, как и в случае стирающего канала, может быть достигнута с помощью случайного стабилизирующего кодирования. Теперь, когда Боб измеряет $n - k$ (выбранных случайным образом, коммутирующих) генераторов стабилизатора, он надеется получить синдром, указывающий на единственный из типичных паулиевских операторов ошибок, возникающих с конечной вероятностью, когда деполяризующий канал действует на n отправленных Алисой кубитов. Для любых $\delta, \varepsilon > 0$ и достаточно большого n, количество N_{typ} типичных операторов Паули с полной вероятностью $1 - \varepsilon$ можно ограничить

следующим образом:

$$N_{\text{typ}} \leqslant 2^{n(H_2(p)+p\log_2 3+\delta)}. \tag{7.259}$$

Попытка восстановления может оказаться неудачной, если среди этих типичных операторов Паули существует еще хотя бы один, имеющий тот же синдром, что и фактический оператор ошибки. Поскольку случайный код сопоставляет случайный $(n-k)$-битовый синдром для каждого оператора Паули, вероятность сбоя можно ограничить неравенством

$$P_{\text{fail}} \leqslant 2^{n(H_2(p)+p\log_2 3+\delta)}2^{k-n} + \varepsilon. \tag{7.260}$$

Здесь второй член ограничивает вероятность атипичной ошибки, а первый — вероятность неоднозначного синдрома в случае типичной ошибки. Мы видим, что усредненная по случайным стабилизирующим кодам вероятность сбоя становится сколь угодно малой при больших n, любых $\delta' < 0$ и такой скорости воспроизведения R, что

$$R \equiv \frac{k}{n} < 1 - H_2(p) - p\log_2 3 - \delta'. \tag{7.261}$$

Если усредненная по кодам вероятность сбоя мала, то существует особый код с малой вероятностью сбоя и, следовательно, скорость воспроизведения R достижима; пропускная способность канала деполяризации ограничена снизу неравенством

$$Q(p) \geqslant 1 - H_2(p) - p\log_2 3. \tag{7.262}$$

Не случайно, что достижимая случайным кодированием скорость воспроизведения согласуется с асимптотической формой квантовой верхней границы Хэмминга для скорости воспроизведения невырожденных кодов $[[n,k,d]]$ при $d > 2pn$; к обоим результатам мы приходим, приписывая свой синдром каждой типичной ошибке. Конечно, нижняя граница Гилберта–Варшамова для скорости воспроизведения кодов $[[n,k,d]]$ лежит ниже $Q(p)$, поскольку она получена при условии, что код может исправлять не только типичные, но и *все* ошибки с весом, не превышающим pn.

Это доказательство методом случайного кодирования можно также применить к несколько более общему каналу, в котором возможны ошибки $\mathbf{X}$, $\mathbf{Y}$ и $\mathbf{Z}$, возникающие с различными частотами. (Назовем его «каналом Паули».) Если ошибка $\mathbf{X}$ возникает с вероятностью p_X, ошибка $\mathbf{Y}$ — с вероятностью p_Y, ошибка $\mathbf{Z}$ — с вероятностью p_Z, а с вероятностью $p_I = 1 - p_X - p_Y - p_Z$ не возникает никакой ошибки, то количество типичных

ошибок в n кубитах равно

$$\frac{n!}{(p_X n)!(p_Y n)!(p_Z n)!(p_I n)!} \sim 2^{nH(p_I,p_X,p_Y,p_Z)}, \tag{7.263}$$

где

$$H \equiv H(p_I,p_X,p_Y,p_Z) = \\ = -p_I \log_2 p_I - p_X \log_2 p_X - p_Y \log_2 p_Y - p_Z \log_2 p_Z \tag{7.264}$$

— энтропия Шеннона распределения вероятностей $\{p_I,p_X,p_Y,p_Z\}$. Теперь мы находим

$$Q(p_I,p_X,p_Y,p_Z) \geqslant 1 - H(p_I,p_X,p_Y,p_Z); \tag{7.265}$$

если скорость воспроизведения R удовлетворяет неравенству $R < 1 - H$, тогда снова крайне маловероятно, что отдельный синдром случайного стабилизирующего кода укажет более, чем на один оператор типичной ошибки.

7.16.3. Вырождение и пропускная способность

Наш вывод нижней границы пропускной способности деполяризующего канала имеет близкое сходство с приведенным в разделе 5.1.3 выводом нижней границы пропускной способности классического двоичного симметричного канала. В классическом случае существует согласованная верхняя граница. Если бы скорость воспроизведения была больше, тогда не было бы достаточного количества синдромов, присваиваемых всем типичным ошибкам.

В квантовом случае это рассуждение не проходит, поскольку квантовые коды могут быть вырожденными. Мы не можем требовать существования своего синдрома у каждой типичной ошибки, так как действие некоторых из них в кодовом пространстве может быть тривиальным. Справедливость теряет не только сам вывод; верхняя самосогласованная граница действительно не существует, то есть в квантовом случае *достижимы* скорости воспроизведения, превышающие $1 - H_2(p) - p \log_2 3$.[1]

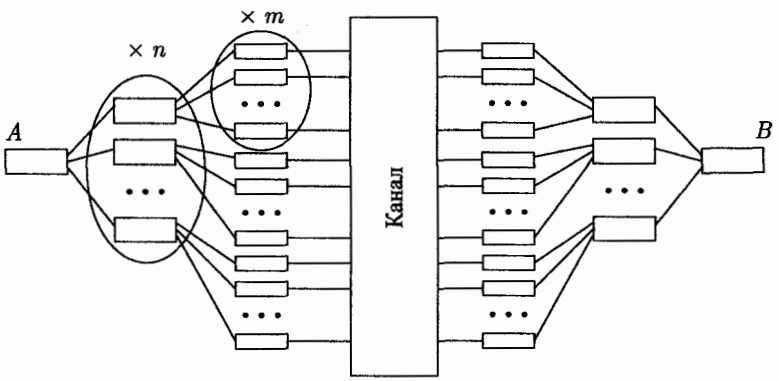
Шор и Смолин исследовали скорость воспроизведения, которая может быть достигнута с помощью каскадного кода, состоящего из случайного

[1] P. M. Shor and J. A. Stmolin, *Quantum Error-Correcting Codes Need Not Completely Reveal the Error Syndrome*, quant-ph/9604006; D. P. DiVincenzo, P. W. Shor, and J. A. Smolin, *Quantum Channel Capacity of Very Noisy Channels*, Phys. Rev. **A57**, pp. 830–839 (1998); quant-ph/9706061.

стабилизирующего кода в качестве внешнего и вырожденного кода с относительно малым размером блока в качестве внутреннего. Согласно их идее, вырождение внутреннего кода позволяет достаточному количеству ошибок действовать тривиально в кодовом пространстве, благодаря чему может быть превышена скорость воспроизведения, достигаемая посредством одного лишь случайного кодирования.

Чтобы изучить эту схему, представим, что как кодирование, так и декодирование выполняются в два этапа. На первом этапе Алиса кодирует выбранное ей состояние в большом n-кубитовом блоке, используя согласованный с Бобом (случайный) внешний код. На втором этапе Алиса кодирует каждый из этих n кубитов в блоке из m кубитов, используя внутренний код. Подобным образом, когда Боб получает nm кубитов, он сначала декодирует каждый внутренний блок из m, а затем — блок из n кубитов.

Очевидно, эту процедуру можно описать на альтернативном языке: Алиса и Боб используют лишь внешний код, но кубиты передаются по составному каналу:



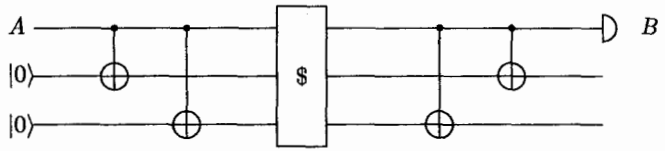
Этот модифицированный канал включает в себя (как показано на рисунке): во-первых, внутреннее кодирование, во-вторых, распространение по исходному каналу с помехами и, наконец, внутреннее декодирование и внутреннее восстановление. Скорость воспроизведения, которая в исходном канале может быть достигнута с помощью каскадного кодирования, такая же, как и скорость, которая может быть достигнута с помощью случайного кодирования в модифицированном канале.

В частности, предположим, что внутренним кодом является m-кубитовый код повторения со стабилизатором

$$\mathbf{Z}_1\mathbf{Z}_2, \quad \mathbf{Z}_1\mathbf{Z}_3, \quad \mathbf{Z}_1\mathbf{Z}_4, \ \ldots, \ \mathbf{Z}_1\mathbf{Z}_m. \tag{7.266}$$

Это далеко не лучший квантовый код; он имеет единичное расстояние, так как нечувствителен к фазовым ошибкам — каждый оператор $\mathbf{Z}_j$ коммутирует со стабилизатором. Но в данном случае для нас важнее высокая высокая степень его вырождения, все ошибки $\mathbf{Z}_i$ эквивалентны.

Кодирующая (и декодирующая) схема для кода повторения состоит лишь из $m-1$ вентилей CNOT, так что наш составной канал выглядит следующим образом (в случае $m=3$):



(Здесь не изображен заключительный восстановительный этап декодирования; например, если оба измеренных кубита показывают 1, то следует инвертировать информационный кубит. В действительности, чтобы упростить исследование составного канала, мы пренебрегаем этим шагом.)

Поскольку CNOT распространяет инвертирование вперед (от управляющего кубита к цели), а обращение фазы назад (от цели к управляющему кубиту), нетрудно понять, что для каждого возможного результата измерения вспомогательных кубитов составной канал является каналом Паули. Представим, что это измерение $m-1$ кубитов внутреннего блока выполняется для каждого из n кубитов внешнего блока. Тогда на каждый из n кубитов действует независимый канал Паули, характеризуемый своим набором параметров (вероятностей ошибок $p_I^{(i)}, p_X^{(i)}, p_Y^{(i)}, p_Z^{(i)}$ для i-го кубита). Таким образом, количество действующих на n кубитов операторов типичных ошибок равно

$$2^{\sum_{i=1}^{n} H_i}, \tag{7.267}$$

где

$$H_i = H(p_I^{(i)}, p_X^{(i)}, p_Y^{(i)}, p_Z^{(i)}) \tag{7.268}$$

— энтропия Шеннона канала Паули, действующего на i-й кубит. Согласно закону больших чисел, для большого n мы получим

$$\sum_{i=1}^{n} H_i = n\langle H\rangle, \tag{7.269}$$

где $\langle H\rangle$ — энтропия Шеннона, усредненная по 2^{m-1} возможным классическим результатам измерения дополнительных кубитов внутреннего кода.

Следовательно, скорость воспроизведения, которая может быть достигнута с помощью случайного внешнего кода, равна

$$R = \frac{1 - \langle H \rangle}{m} \tag{7.270}$$

(мы делим ее на m, потому что каскадный код имеет длину, в m раз большую, чем случайный код).

Шор и Смолин обнаружили, что существуют (m-кратные) коды повторения, для которых (в подходящем диапазоне p) $1 - \langle H \rangle$ является положительной величиной, тогда как $1 - H_2(p) - p \log_2 3$ — отрицательной. Тогда, в этом диапазоне, пропускная способность $Q(p)$ ненулевая, следовательно, нижняя граница (7.262) не является строгой.

Асимптотически неисчезающая скорость воспроизведения достижима посредством случайного кодирования при $1 - H_2(p) - p \log_2 3 > 0$, или $p < p_{\max} \simeq 0{,}18929$. Если случайный внешний код каскадируется с 5-кубитовым внутренним кодом повторения ($m = 5$ оказывается оптимальным выбором), тогда $1 - \langle H \rangle > 0$ при $p < p'_{\max} \simeq 0{,}19036$; максимальная вероятность ошибки, для которой достижима ненулевая скорость воспроизведения, возрастает примерно на 0,6%. То, что в этом диапазоне вероятностей ошибки каскадный код должен превзойти случайный, не является очевидным, хотя, как мы отметили, этого можно было ожидать, из-за (фазового) вырождения кода повторения. Не очевидно также и то, что $m = 5$ должно быть лучшим выбором, но это можно проверить явным вычислением $\langle H \rangle$.[1]

Деполяризующий канал является одним из наиболее простых квантовых каналов. Но даже для этого случая проблема характеристики и вычисления пропускной способности во многом не решена. Этот пример показывает, что из-за возможности вырожденного кодирования проблема пропускной способности для квантовых каналов оказывается куда более острой, чем для классических каналов.

Мы видели, что (если ошибки хорошо описываются деполяризующим каналом) квантовую информацию можно извлечь из квантовой памяти со сколь угодно высокой точностью воспроизведения, пока вероятность ошибки на один кубит меньше 19%. Это является улучшением относительно 10%-ой частоты появления ошибок, которой, как мы обнаружили, можно пользоваться при каскадном соединении кода $[[5, 1, 3]]$. В действительности, коды $[[n, k, d]]$, которые могут исправить вплоть до np ошибок любо-

[1] На самом деле можно достичь дальнейшего очень незначительного улучшения путем каскадного соединения случайного кода с описанным в упражнениях 25-кубитовым обобщенным кодом Шора — тогда ненулевая скорость достигается при $p < p'_{\max} \simeq 0{,}19056$ (еще на 0.1% лучше максимально допустимой вероятности ошибки в коде повторения).

го распределения, согласно границе Рейнса не существуют при $p > 1/6$. Ненулевая пропускная способность возможна для частот появления ошибок в интервале от 16.7% до 19%, потому что для ККК0 достаточно быть способным исправлять типичные ошибки, а не все возможные.

Однако утверждение о том, что восстановление возможно, даже если 19% кубитов подвергаются разрушению, весьма обманчиво в одном важном отношении. Этот результат применим, если кодирование, декодирование и восстановление могут быть выполнены безупречно. Но эти операции на самом деле представляют собой очень сложные квантовые вычисления, которые на практике, конечно, будут чувствительны к ошибкам. Мы не сможем полностью понять, насколько хорошо кодирование может защитить квантовую информацию от повреждений, пока не научимся составлять протокол исправления ошибок, надежный, даже если исполнение самого протокола неидеально. Такие помехоустойчивые протоколы обсуждаются в приложении.

7.17. Итоги

Квантовые коды коррекции ошибок. Коррекция квантовых ошибок может защитить квантовую информацию от декогерентизации и «унитарных ошибок», возникающих вследствие неидеальной реализации квантовых вентилей. В (двоичном) *квантовом коде коррекции ошибок* (ККК0) 2^k-мерное гильбертово пространство k закодированных кубитов $\mathcal{H}_{\text{code}}$ вложено в 2^n-мерное гильбертово пространство n кубитов. Действующие на n кубитов ошибки обратимы при условии, что $\langle\psi|\mathbf{M}_\nu^\dagger\mathbf{M}_\mu|\psi\rangle/\langle\psi|\psi\rangle$ не зависит от $|\psi\rangle$ для любого $|\psi\rangle \in \mathcal{H}_{\text{code}}$ и любых двух операторов Крауса $\mathbf{M}_{\mu,\nu}$, возникающих в разложении супероператора ошибки. Супероператор восстановления преобразует запутывание окружения с кодовым блоком в запутывание окружения со служебным кубитом, который затем может быть выброшен.

Квантовые стабилизирующие коды. Большинство ККК0, которые могут быть построены, представляют собой *стабилизирующие коды*. Двоичный стабилизирующий код характеризуется своим стабилизатором S и абелевой подгруппой n-кубитовой *группы Паули* $G_n = \{\mathbf{1}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}^{\otimes n}$ (где $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ — однокубитовые операторы Паули). Кодовое подпространство представляет собой пространство состояний, одновременно являющихся собственными векторами всех элементов S, соответствующими единичному собственному значению; если S имеет $n-k$ независимых генераторов, тогда существует k закодированных кубитов. Стабилизирующий код может исправить каждую ошибку из подмножества $\mathcal{E}$ группы G_n, если

для каждого $\mathbf{E}_a, \mathbf{E}_b \in \mathcal{E}$ оператор $\mathbf{E}_a^\dagger \mathbf{E}_b$ или принадлежит стабилизатору S, или не принадлежит нормализатору стабилизатора $S^\perp$. Если для $\mathbf{E}_{a,b} \in \mathcal{E}$ некоторый оператор $\mathbf{E}_a^\dagger \mathbf{E}_b$ принадлежит S, то код *вырожден*; в противном случае — *невырожден*. Операторы из $S^\perp \setminus S$ представляют собой «логические» операторы, действующие на закодированную квантовую информацию. Стабилизатор S может быть связан с аддитивным кодом над конечным полем $\mathrm{GF}(4)$, самоортогональным относительно симплектического внутреннего произведения. *Весом* оператора Паули является количество кубитов, на которые он действует нетривиально, а *расстоянием* d стабилизирующего кода — минимальный вес элемента из $S^\perp \setminus S$. Код, имеющий длину n, k закодированных кубитов и расстояние d, называется квантовым кодом $[[n, k, d]]$. Если код осуществляет восстановление от любого супероператора ошибки с носителем на операторах Паули с весами, не превышающими t, то мы говорим, что код «может исправить t ошибок». Код с расстоянием d может исправить $(d-1)/2$ ошибок в неизвестных позициях и $d-1$ ошибок в известных позициях. Можно построить «хорошие» семейства стабилизирующих кодов, в которых d/n и k/n остаются отличными от нуля при $n \to \infty$.

Примеры. Квантовый код $[[5, 1, 3]]$, связанный с классическим кодом Хэмминга над $\mathrm{GF}(4)$, представляет собой код минимальной длины, способный исправить одну ошибку. По заданному классическому линейному коду C_1 и его субкоду $C_2 \subseteq C_1$ можно построить квантовый код Колдербенка–Шора–Стина (КШС-код) с $k = \dim(C_1) - \dim(C_2)$ закодированными кубитами. Расстояние d КШС-кода удовлетворяет неравенству $d \geqslant \min(d_1, d_2^\perp)$, где d_1 — расстояние C_1, а $d_2^\perp$ — расстояние $C_2^\perp$, дуального коду C_2. Простейшим КШС-кодом является квантовый код $[[7, 1, 3]]$, построенный из классического кода Хэмминга $[7, 4, 3]$ и его четного субкода. *Каскадное соединение* квантовых кодов $[[n_1, 1, d_1]]$ и $[[n_2, 1, d_2]]$ дает вырожденный код $[[n_1 n_2, 1, d]]$ с $d \geqslant d_1 d_2$.

Пропускная способность квантового канала. Пропускной способностью квантового канала (квантового канала с помехами) является максимальная скорость воспроизведения, с которой квантовая информация может быть передана по каналу и декодирована со сколь угодно высокой точностью воспроизведения. Пропускная способность двоичного стирающего канала с вероятностью стирания p равна $Q(p) = 1 - 2p$ при $0 \leqslant p \leqslant 1/2$. Пропускная способность двоичного деполяризующего канала до сих пор неизвестна. Ее вычисление представляет собой довольно тонкую проблему, поскольку оптимальный код может быть вырожденным; в частности, случайные коды не достигают асимптотически оптимальной скорости воспроизведения по квантовому каналу.

7.18. Упражнения

7.1. Код коррекции фазовых ошибок. a) Постройте генераторы стабилизатора для кода с $n = 3$, $k = 1$, который может исправить одно инвертирование бита; убедитесь в том, что восстановление возможно при любой ошибке из множества $\mathcal{E} = \{\mathbf{111}, \mathbf{X11}, \mathbf{1X1}, \mathbf{11X}\}$. Найдите ортонормированный базис для двумерного кодового подпространства.

b) Постройте генераторы стабилизатора для кода с $n = 3$, $k = 1$, который может исправить одну фазовую ошибку; убедитесь в том, что восстановление возможно при любой ошибке из множества $\mathcal{E} = \{\mathbf{111}, \mathbf{Z11}, \mathbf{1Z1}, \mathbf{11Z}\}$. Найдите ортонормированный базис для двумерного кодового подпространства.

7.2. Коды обнаружения ошибок. a) Постройте генераторы стабилизатора для квантового кода $[[n, k, d]] = [[3, 0, 2]]$. Используя этот код, мы можем обнаружить любую однокубитовую ошибку. Найдите закодированное состояние. (Не кажется ли оно вам знакомым?)

b) Два ККО C_1 и C_2 (с одинаковой длиной n) *эквивалентны*, если перестановка кубитов, в совокупности с однокубитовым унитарным преобразованием, превращает кодовое подпространство C_1 в подпространство C_2. Все ли стабилизирующие коды $[[3, 0, 2]]$ эквивалентны?

c) Существует ли стабилизирующий код $[[3, 1, 2]]$?

7.3. Максимальное запутывание. Рассмотрите квантовый код $[[5, 1, 3]]$, генераторы стабилизатора которого $\mathbf{M}_1 = \mathbf{XZZX1}$, а $\mathbf{M}_{2,3,4}$ получаются из $\mathbf{M}_1$ циклическими перестановками, и выберите в качестве закодированной операции $\bar{\mathbf{Z}} = \mathbf{ZZZZZ}$. Из закодированных состояний $|\bar{0}\rangle$ с $\bar{\mathbf{Z}}|\bar{0}\rangle = |\bar{0}\rangle$ и $|\bar{1}\rangle$ с $\bar{\mathbf{Z}}|\bar{1}\rangle = -|\bar{1}\rangle$ постройте код с $n = 6$, $k = 0$, закодированное состояние которого имеет вид

$$\frac{1}{\sqrt{2}}(|0\rangle \otimes |\bar{0}\rangle + |1\rangle \otimes |\bar{1}\rangle). \tag{7.271}$$

a) Постройте множество генераторов стабилизатора для этого кода с $n = 6$, $k = 0$.

b) Определите расстояние этого кода. (Вспомните, что для кода с $k = 0$ расстояние определяется как минимальный вес любого элемента стабилизатора.)

c) Найдите $\boldsymbol{\rho}^{(3)}$, матрицу плотности, которая получается, если выбираются три кубита, а по состояниям трех других кубитов вычисляется след.

7.4. Кодовые слова и нелокальность. Для кода $[[5,1,3]]$ с генераторами стабилизатора и логическими операторами из предыдущей задачи:

a) Выразите $\bar{\mathbf{Z}}$ как оператор Паули с весом три, через тензорное произведение операторов $\mathbf{1}$, $\mathbf{X}$ и $\mathbf{Z}$ (без $\mathbf{Y}$). Обратите внимание, что вследствие цикличности кода все циклические перестановки вашего выражения являются эквивалентными способами представления $\bar{\mathbf{Z}}$.

b) Используйте предположение об эйнштейновской локальности (скрытые локальные переменные), чтобы предсказать зависимость между пятью (связанными циклически) найденными в **(a)** наблюдаемыми и наблюдаемой $\mathbf{ZZZZZ}$. Выполняется ли эта связь между наблюдаемыми в состоянии $|\bar{0}\rangle$?

c) Что сказал бы об этом Эйнштейн?

7.5. Обобщенный код Шора. Для целого числа $m \geqslant 2$, рассмотрите обобщение 9-кубитового кода Шора с параметрами $n = m^2$, $k = 1$ и кодовым подпространством, натянутым на два состояния:

$$\begin{aligned} |\bar{0}\rangle &= (|000\ldots0\rangle + |111\ldots1\rangle)^{\otimes m}, \\ |\bar{1}\rangle &= (|000\ldots0\rangle - |111\ldots1\rangle)^{\otimes m}. \end{aligned} \tag{7.272}$$

a) Постройте генераторы стабилизатора для этого кода, а так же логические операции $\bar{\mathbf{Z}}$ и $\bar{\mathbf{X}}$, такие что

$$\begin{aligned} \bar{\mathbf{Z}}|\bar{0}\rangle &= |\bar{0}\rangle, & \bar{\mathbf{X}}|\bar{0}\rangle &= |\bar{1}\rangle, \\ \bar{\mathbf{Z}}|\bar{1}\rangle &= -|\bar{1}\rangle, & \bar{\mathbf{X}}|\bar{1}\rangle &= |\bar{0}\rangle. \end{aligned} \tag{7.273}$$

b) Каково расстояние этого кода?

c) Предположите, что m — нечетное число и что каждый из $n = m^2$ кубитов подвергается действию деполяризующего канала с вероятностью ошибки p. Насколько хорошо этот код защищает закодированный кубит? В частности,

(i) в главном нетривиальном порядке по p, оцените вероятность логической ошибки инвертирования бита $|\bar{0}\rangle \leftrightarrow |\bar{1}\rangle$,

(ii) в главном нетривиальном порядке по p, оцените вероятность логической фазовой ошибки $|\bar{0}\rangle \to |\bar{0}\rangle$, $|\bar{1}\rangle \to -|\bar{1}\rangle$.

d) Рассмотрите асимптотическое поведение вашего результата в **(c)** при большом m. Какому условию должно удовлетворять p, чтобы код обеспечил хорошую защиту: (i) от инвертирования битов и (ii) от фазовых ошибок в пределе $n \to \infty$?

7.6. Кодирующие схемы. Для квантового кода $[[n, k, d]]$ кодирующим преобразованием служит унитарное преобразование $\mathbf{U}$, действующее как

$$\mathbf{U} : |\psi\rangle \otimes |0\rangle^{\otimes(n-k)} \rightarrow |\bar{\psi}\rangle, \tag{7.274}$$

где $|\psi\rangle$ — произвольное k-кубитовое состояние, а $|\bar{\psi}\rangle$ — соответствующее закодированное состояние. Разработайте квантовую схему, осуществляющую кодирующее преобразование для

a) кода Шора $[[9, 1, 3]]$;

b) кода Стина $[[7, 1, 3]]$.

7.7. Укорачивание квантового кода. a) Рассмотрите двоичный стабилизирующий код $[[n, k, d]]$. Покажите, что можно выбрать $n - k$ генераторов стабилизатора так, чтобы на последний кубит нетривиально действовали не более двух (то есть оставшиеся $n - k - 2$ генераторов применяют к последнему кубиту 1).

b) Эти $n - k - 2$ генераторов стабилизатора, применяющих 1 к последнему кубиту, по-прежнему будут коммутирующими и независимыми, если мы выбросим последний кубит. Следовательно, они представляют собой генераторы для кода с длиной $n-1$ и $k+1$ закодированными кубитами. Покажите, что если исходный код невырожден, то расстояние укороченного кода равно по крайней мере $d - 1$. (**Указание:** Сначала покажите, что если существует элемент $(n - 1)$-кубитовой группы Паули с весом t, коммутирующий со стабилизатором укороченного кода, то существует элемент n-кубитовой группы Паули с весом, не превышающим $t + 1$, коммутирующий со стабилизатором исходного кода.)

c) Примените процедуру укорачивания **(a)** и **(b)** для ККК $[[5, 1, 3]]$. Узнаёте ли вы полученный код? (**Указание:** Может оказаться полезным воспользоваться свободой выбора базиса для некоторых из кубитов.)

7.8. Коды для кудитов. *Кудит* представляет собой d-мерную квантовую систему. Действующие на кубиты операторы Паули $\mathbf{I}$, $\mathbf{X}$ и $\mathbf{Z}$ можно обобщить на кудиты следующим образом. Пусть $\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\}$ обозначает ортонормированный базис в гильбертовом пространстве одного кудита. Определите операторы:

$$\begin{aligned} \mathbf{X} &: |j\rangle \rightarrow |j + 1 \,(\mathrm{mod}\, d)\rangle, \\ \mathbf{Z} &: |j\rangle \rightarrow \omega^j |j\rangle, \end{aligned} \tag{7.275}$$

где $\omega = \exp(2\pi i/d)$. Тогда $d \times d$-операторы Паули $\mathbf{E}_{r,s}$ равны

$$\mathbf{E}_{r,s} = \mathbf{X}^r \mathbf{Z}^s, \quad r, s = 0, 1, \ldots, d-1. \tag{7.276}$$

a) Образуют ли $\mathbf{E}_{r,s}$ базис в пространстве операторов, действующих на кубит? Унитарны ли они? Вычислите $\mathrm{tr}(\mathbf{E}_{r,s}^\dagger \mathbf{E}_{t,u})$.

b) Операторы Паули удовлетворяют условиям

$$\mathbf{E}_{r,s}\mathbf{E}_{t,u} = \eta_{r,s;t,u}\mathbf{E}_{t,u}\mathbf{E}_{t,u}, \tag{7.277}$$

где $\eta_{r,s;t,u}$ — фазовый множитель. Вычислите этот фазовый множитель. n-кратные тензорные произведения этих действующих на кудит операторов Паули образуют группу $G_n^{(d)}$ порядка d^{2n+1} (и если мы удалим его d-элементный центр, то получим группу $\bar{G}_n^{(d)}$ порядка d^{2n}). Чтобы построить стабилизирующий код для кудитов, мы выбираем абелеву подгруппу группы $G_n^{(d)}$ с $n-k$ генераторами; такой код является общим собственным состоянием этих генераторов с собственным значением единица. Если d — простое число, то кодовое подпространство имеет размерность d^k: k логических кудитов закодированы в блоке из n кудитов.

c) Объясните, насколько иной может быть размерность, если d не является простым числом. (**Указание:** Рассмотрите случай $d = 4$ и $n = 1$.)

7.9. Измерение синдрома для кудитов. Ошибки в кудитах выявляются при измерении генераторов стабилизатора. С этой целью мы можем осуществить двухкудитовый вентиль SUM (который обобщает вентиль CNOT), действующий как

$$\mathrm{SUM} : |j\rangle \otimes |k\rangle \rightarrow |j\rangle \otimes |k + j \,(\mathrm{mod}\ d)\rangle. \tag{7.278}$$

a) Опишите содержащую вентили SUM квантовую схему, которую можно осуществить для измерения n-кудитовой наблюдаемой вида

$$\bigotimes_a \mathbf{Z}_a^{s_a}. \tag{7.279}$$

Если d — простое число, то для каждого $r, s = 0, 1, 2, \ldots, d-1$ существует такой однокудитовый унитарный оператор $\mathbf{U}_{r,s}$, что

$$\mathbf{U}_{r,s}\mathbf{E}_{r,s}\mathbf{U}_{r,s}^\dagger = \mathbf{Z}. \tag{7.280}$$

b) Опишите содержащую вентили SUM и $\mathbf{U}_{r,s}$ квантовую схему, которую можно осуществить для измерения произвольного элемента $G_n^{(d)}$ вида

$$\bigotimes_a \mathbf{E}_{r_a,s_a}. \tag{7.281}$$

7.10. Коды обнаружения ошибок для кудитов. Кудит с $d = 3$ называется *кутритом*. Рассмотрите кутритовый стабилизирующий код с длиной $n = 3$ и с одним ($k = 1$) закодированным кутритом, определяемый двумя генераторами стабилизатора

$$\mathbf{ZZZ}, \quad \mathbf{XXX}. \tag{7.282}$$

a) Коммутируют ли генераторы?

b) Определите расстояние кода.

c) Найдите явное выражение ортонормированного базиса для трехмерного кодового подпространства через ортонормированный базис $\{|0\rangle, |1\rangle, |2\rangle\}$ для кутрита.

d) Постройте генераторы стабилизатора для $n = 3m$ кутритового кода (где m — произвольное положительное целое число) с $k = n - 2$, который может обнаружить одну ошибку.

e) Постройте генераторы стабилизатора для выявляющего одну ошибку кудитового кода с параметрами $n = d$, $k = d - 2$.

7.11. Коды коррекции ошибок для кудитов. Рассмотрите кудитовый стабилизирующий код при $n = 5$, $k = 1$ с генераторами стабилизатора

$$\begin{array}{lccccc} \mathbf{M}_1 = & \mathbf{X} & \mathbf{Z} & \mathbf{Z}^{-1} & \mathbf{X}^{-1} & \mathbf{1} \\ \mathbf{M}_2 = & \mathbf{1} & \mathbf{X} & \mathbf{Z} & \mathbf{Z}^{-1} & \mathbf{X}^{-1} \\ \mathbf{M}_3 = & \mathbf{X}^{-1} & \mathbf{1} & \mathbf{X} & \mathbf{Z} & \mathbf{Z}^{-1} \\ \mathbf{M}_4 = & \mathbf{Z}^{-1} & \mathbf{X}^{-1} & \mathbf{1} & \mathbf{X} & \mathbf{Z} \end{array} \tag{7.283}$$

(второй, третий и четвертый генераторы получены из первого при помощи циклических перестановок кудитов).

a) Определите порядок каждого генератора. Действительно ли генераторы независимы? Коммутируют ли они? Является ли пятая циклическая перестановка $\mathbf{ZZ}^{-1}\mathbf{X}^{-1}\mathbf{1X}$ независимой от остальных?

b) Определите расстояние этого кода. Является ли код невырожденным?

c) Постройте закодированные операции $\bar{\mathbf{X}}$ и $\bar{\mathbf{Z}}$, выразив их как операторы с весом три. (Убедитесь в том, что для любого значения d эти операторы подчиняются правильным коммутационным соотношениям).

Решения упражнений к главе 7[1]

7.1. Коды коррекции фазовых ошибок

a) Квантовый код коррекции инвертирования бита является классическим кодом повторения. Он имеет генераторы стабилизатора

$$\mathbf{M}_1 = \mathbf{ZZ1},$$
$$\mathbf{M}_2 = \mathbf{1ZZ}.$$

Выбор представления закодированных операторов в виде

$$\bar{\mathbf{X}} = \mathbf{XXX},$$
$$\bar{\mathbf{Z}} = \mathbf{ZZZ}$$

отбирает базис кодовых слов

$$|\bar{0}\rangle = |000\rangle,$$
$$|\bar{1}\rangle = |111\rangle.$$

Другое представление закодированных операторов привело бы к другому базису. Данный выбор делает этот код наиболее похожим на его классический аналог.

b) Квантовый код коррекции обращения фазы также представляет собой классический код повторения, хотя и в другом базисе. Он имеет генераторы стабилизатора

$$\mathbf{M}_1 = \mathbf{XX1},$$
$$\mathbf{M}_2 = \mathbf{1XX}.$$

Выбор такого же, как и выше, представления закодированных операторов

$$\bar{\mathbf{X}} = \mathbf{XXX},$$
$$\bar{\mathbf{Z}} = \mathbf{ZZZ}$$

отбирает базис кодовых слов

$$|\bar{0}\rangle = \frac{1}{\sqrt{8}}(|0\rangle + |1\rangle)^{\otimes 3},$$

$$|\bar{1}\rangle = \frac{1}{\sqrt{8}}(|0\rangle - |1\rangle)^{\otimes 3}.$$

[1] Решения выполнены Эндрю Лэндалом.

Заметим, что если мы локально совершим адамаровский поворот базиса каждого кубита [$\mathbf{H}|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, $\mathbf{H}|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$] и операторов стабилизатора [$\mathbf{HZH}^{-1} = \mathbf{X}$, $\mathbf{HXH}^{-1} = \mathbf{Z}$], то получим тот же самый код, что и в части **(a)**. По этой причине можно сказать, что квантовые коды инвертирования бита и обращения фазы *эквивалентны с точностью до локальных унитарных преобразований* [ср. задачу **7.2(b)**].

7.2. Коды детектирования ошибок

a) Код $[[3, 0, 2]]$ можно построить, дополняя один из кодов из задачи **7.1** закодированной операцией (коммутирующей со стабилизатором). Чтобы сохранить расстояние кода, эту закодированную операцию следует выбрать с минимальным весом не ниже двух. [Имея в виду то, как в задаче **7.3(b)** определяется расстояние кода с $k = 0$.] Одним из таких выборов является

$$\begin{aligned}\mathbf{M}_1 &= \mathbf{ZZ1},\\ \mathbf{M}_2 &= \mathbf{1ZZ},\\ \mathbf{M}_3 &= \mathbf{XXX}.\end{aligned}$$

Закодированным состоянием [то есть $\bar{\mathbf{X}}|\bar{0}\rangle$ из **7.1(a)**] является знакомое трехкубитовое кот-состояние, также известное как состояние Гринбергера–Горна–Цайлингера (ГГЦ):

$$\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle).$$

b) Да, в этом смысле все стабилизирующие коды $[[3, 0, 2]]$ эквивалентны. Покажем, что это так, различными способами конструируя их генераторы стабилизаторов. Такой подход ведет к ответу, который больше похож на решение «логической загадки».

Начнем с того, что стабилизатор содержит всего $n - k = 3$ генератора, которые в самом общем случае имеют вид

$$\begin{aligned}\mathbf{M}_1 &= \pm\mathbf{ABC},\\ \mathbf{M}_2 &= \pm\mathbf{DEF},\\ \mathbf{M}_3 &= \pm\mathbf{GHJ}.\end{aligned}$$

С помощью локальных унитарных преобразований (ЛУП) всегда можно избавиться от общих фаз генераторов.

Код должен антикоммутировать со всеми ошибками единичного веса (чтобы детектировать их!), то есть $\mathbf{A}$, $\mathbf{D}$ и $\mathbf{G}$ не могут быть все одинаковыми. Следовательно, без потери общности можно считать $\mathbf{A} \neq \mathbf{D}$, более того, используя ЛУП, преобразовать $\mathbf{A} \to \mathbf{X}$, $\mathbf{D} \to \mathbf{Z}$. Тогда, действуя на $\mathbf{M}_3$ операторами $\mathbf{M}_1$ или $\mathbf{M}_2$ (или обоими) до тех пор пока $\mathbf{G}$ не обратится в единичный $\mathbf{1}$, получим

$$\begin{aligned}\mathbf{M}_1 &= \mathbf{XBC},\\ \mathbf{M}_2 &= \mathbf{ZEF},\\ \mathbf{M}_3 &= \mathbf{1HJ}.\end{aligned}$$

Все генераторы должны коммутировать, следовательно либо $\{\mathbf{B},\mathbf{E}\} = 0$ либо $\{\mathbf{C},\mathbf{F}\} = 0$. Вновь без потери общности (при необходимости поменяв местами второй и третий кубиты) можно положить $\{\mathbf{B},\mathbf{E}\} = 0$ и, применяя ЛУП, выбрать $\mathbf{B} = \mathbf{X}$ и $\mathbf{E} = \mathbf{Z}$. Теперь генераторы выглядят как

$$\begin{aligned}\mathbf{M}_1 &= \mathbf{XXC},\\ \mathbf{M}_2 &= \mathbf{ZZF},\\ \mathbf{M}_3 &= \mathbf{1HJ}.\end{aligned}$$

Так как ошибка с единичным весом $\mathbf{11J}$ не должна коммутировать со стабилизатором, операторы $\mathbf{C}$ и $\mathbf{F}$ не могут одновременно быть равными единичному $\mathbf{1}$. Как мы увидим в задаче **7.3(b)** код с расстоянием два и $k=0$ не может иметь элементов стабилизатора с меньшим, чем два, весом. (В этом смысле все коды с $k = 0$ невырождены.) Следовательно, ошибка $\mathbf{11J}$ не может принадлежать стабилизатору, то есть $\mathbf{H} \neq \mathbf{1}$. Но поскольку $[\mathbf{M}_1, \mathbf{M}_2] = 0$, то $[\mathbf{C},\mathbf{F}] = 0$. Следовательно, или $\mathbf{C} = \mathbf{F} \neq \mathbf{1}$, или только один из них равен единичному оператору. Так как мы всегда можем применить ЛУП $\mathbf{X} \leftrightarrow \mathbf{Z}$ одновременно к первым двум кубитам, то без потери общности можно выбрать оператор $\mathbf{C}$ не равным единичному. Подействовав другим ЛУПом на третий кубит, можно положить $\mathbf{C} = \mathbf{X}$. Следовательно, оператор $\mathbf{F}$ равен или $\mathbf{X}$, или $\mathbf{1}$. Если $\mathbf{F} = \mathbf{X}$, то, выполняя отображение $\mathbf{M}_2 \to \mathbf{M}_1\mathbf{M}_2 = \mathbf{YY1}$, а затем применяя к первым двум кубитам ЛУП $\mathbf{Y} \leftrightarrow \mathbf{Z}$, второй генератор $\mathbf{M}_2$ можно преобразовать в $\mathbf{ZZ1}$. Следовательно, не теряя общности, можно выбрать $\mathbf{F} = \mathbf{1}$. Теперь генераторы выглядят как

$$\begin{aligned}\mathbf{M}_1 &= \mathbf{XXX},\\ \mathbf{M}_2 &= \mathbf{ZZ1},\\ \mathbf{M}_3 &= \mathbf{1HJ}.\end{aligned}$$

Наконец, из равенств $[\mathbf{M}_2, \mathbf{M}_3] = 0$ и $[\mathbf{M}_1, \mathbf{M}_3] = 0$ следует, что $\mathbf{H} = \mathbf{Z}$, а $\mathbf{J}$ равен или $\mathbf{Z}$, или $\mathbf{1}$. Но $\mathbf{J}$ не может быть равным $\mathbf{1}$, поскольку тогда ошибка с единичным весом $\mathbf{1Z1}$ коммутировала бы с $\mathbf{M}_3$ (фактически, в этом случае $\mathbf{1Z1}$ *совпадает* с $\mathbf{M}_3$). Следовательно, $\mathbf{J} = \mathbf{Z}$ и, значит, наиболее общий стабилизирующий код $[[3, 0, 2]]$ имеет генераторы

$$\mathbf{M}_1 = \mathbf{XXX},$$
$$\mathbf{M}_2 = \mathbf{ZZ1},$$
$$\mathbf{M}_3 = \mathbf{1ZZ}.$$

c) Нет, стабилизирующий код $[[3, 1, 2]]$ не существует. Для того чтобы такой код детектировал все возможные однокубитовые ошибки, каждый столбец генераторов стабилизатора должен содержать как оператор $\mathbf{X}$, так и $\mathbf{Z}$ (или $\mathbf{Y}$). Однако код $[[3, 1, 2]]$ имеет только два генератора стабилизатора, и нет возможности сделать коммутирующими произведение трех пар антикоммутирующих операторов.

Замечание. То, что код $[[3, 1, 2]]$ не существует, может показаться удивительным, поскольку, как мы видели в задаче **7.1(a)**, коды с $n = 3$, $k = 1$, детектирующие ошибку инвертирования бита и обращения фазы, *существуют*. Дело в том, что не существует кода с $n = 3$, $k = 1$, способного детектировать *все* возможные ошибки; такие коды могут только корректировать ошибки, возникающие в некотором *базисе*. Поскольку кубиты весьма дороги, то, стремясь максимизировать эффективность квантового кода коррекции ошибок с $n = 3$, важно знать, в каком базисе предпочитает действовать окружение.

7.3. Максимальное запутывание

a) Пусть $|\psi\rangle$ обозначает закодированное состояние. Мы должны найти $n - k = 6$ линейно независимых фиксирующих $|\psi\rangle$ операторов. Данное в задаче разложение типа Шмидта для $|\psi\rangle$ делает ясным, что это состояние фиксируется всеми операторами вида $\mathbf{1M}$, где $\mathbf{M}$ принадлежит стабилизатору кода $[[5, 1, 3]]$. Более того, поскольку $|\psi\rangle$ имеет ту же природу, что и кот-состояние, то оно фиксируется и операторами $\mathbf{X\bar{X}}$ и $\mathbf{Z\bar{Z}}$. Поэтому полный список генераторов стабилизатора этого кода выглядит следующим образом:

$$\mathbf{M}_1 = \mathbf{1XZZX1},$$

$$\mathbf{M}_2 = \mathbf{11XZZX},$$
$$\mathbf{M}_3 = \mathbf{1X1XZZ},$$
$$\mathbf{M}_4 = \mathbf{1ZX1XZ},$$
$$\mathbf{M}_5 = \mathbf{XXXXXX},$$
$$\mathbf{M}_6 = \mathbf{ZZZZZZ}.$$

b) Расстояние кода равно четырем, и он невырожден. Эти выводы зависят от того, как интерпретируется расстояние и вырождение квантового кода с $k = 0$.

Замечания о расстоянии при $k = 0$

Для кодов с $k = 0$ стандартное понятие расстояния определено недостаточно четко. Обычно мы говорим, что расстоянием кода является наибольший вес представлений минимального веса для закодированных операторов $\bar{\mathbf{X}}$ и $\bar{\mathbf{Z}}$. Но для кодов с $k = 0$ не существует закодированных операторов! Следовательно, мы должны вернуться к самым основам, чтобы понять, что означает расстояние в этом случае.

1) **Расстояние на языке корректирования**

   Один способ состоит в определении расстояния кода с $k = 0$ с помощью его свойств, корректирующих ошибки. Однако это ведет к выводу, что расстояние всегда равно n, так как закодированное состояние *известно*. «Восстановлением» является просто приготовление состояния заново! Эта интерпретация не выглядит достаточно ясной и, фактически, является не лучшим способом думать о том, что здесь происходит.

2) **Расстояние на языке детектирования**

   Физически более мотивированным понятием расстояния является то, которое количественно определяет, сколько из n физических кубитов взаимодействовало с окружением если было приготовлено k логических кубитов. Эта интерпретация фокусируется скорее на *детектировании*, чем на коррекции. Тогда расстояние может быть разумно определено как число, превышающее на единицу максимальное количество детектируемых взаимодействий кубитов ($d = t+1$). Мы можем использовать это определение расстояния, чтобы поставить разумный вопрос о декогерентизирующей силе окружения и экспериментально ответить на него, используя коды с $k = 0$.

   Напомним, что ошибка является детектируемой, если и только если она антикоммутирует с некоторым генератором стабилизатора. Одна-

ко для кодов с $k = 0$ стабилизатору принадлежат только элементы, коммутирующие со *всеми* его же элементами. По этой причине максимальное количество детектируемых ошибок равно минимальному из весов нетривиальных элементов стабилизатора. Это позволяет найти расстояние кода (во второй из приведенных трактовок), лишь посмотрев на его стабилизатор. Воспользуемся этим результатом, чтобы найти расстояние кода, предложенного в части **(а)** этой задачи.

Будем использовать определение (2), чтобы найти расстояние кода $[[6, 0, 4]]$. Поскольку его стабилизатор достаточно мал, мы можем обследовать его явно, чтобы найти элемент с минимальным весом. Его стабилизатором является

$$\begin{array}{rr}
\mathbf{1XZZX1}, & \mathbf{ZY11YZ}, \\
-\mathbf{1YXXY1}, & -\mathbf{ZXYYXZ}, \\
-\mathbf{1ZYYZ1}, & -\mathbf{Z1XX1Z}, \\
\mathbf{111111}, & \mathbf{ZZZZZZ}, \\
 & \\
\mathbf{X1YY1X}, & \mathbf{YZXXZY}, \\
-\mathbf{XZ11ZX}, & -\mathbf{YX11XY}, \\
-\mathbf{XYZZYX}, & -\mathbf{Y1ZZ1Y}, \\
\mathbf{XXXXXX}, & \mathbf{YYYYYY},
\end{array}$$

плюс перестановки.

Непосредственная проверка показывает, что минимальный вес элементов стабилизатора равен четырем.

Замечания о вырождении

В применении к коду с $k = 0$ вырождение также является не четко определенным понятием. Обычно мы говорим, что код вырожден, если минимальный вес элемента стабилизатора меньше d. Но как мы видели выше, для кодов с $k = 0$ расстояние d *определяется* как минимальный вес элементов стабилизатора. В этом смысле все коды с $k = 0$ суть невырожденные.

Более физическим способом определения вырождения кода с $k = 0$ является описание его как вырожденного, если существует две различные детектируемые ошибки, которые одинаково влияют на закодированное состояние. (Следовательно, не существует измерения синдрома, способного

различить эти две ошибки, даже несмотря на то, что можно детектировать, когда происходит одна из них.) Согласно этому определению, код $[[6,0,4]]$ наследует невырожденность своего родительского кода $[[5,1,3]]$.

Используя обе эти интерпретации, мы находим, что квантовый код $[[6,0,4]]$ *невырожден*.

c) В разделе 7.3.4 было в достаточно общем виде доказано, что вычисление следа по $d-1$ кубитам невырожденного кода с расстоянием d дает матрицу плотности, пропорциональную единице. Следовательно,

$$\boldsymbol{\rho}^{(3)} = \frac{1}{8}\mathbf{1}.$$

7.4. Кодовые слова и нелокальность

a) С помощью преобразования $\bar{\mathbf{Z}} \to \bar{\mathbf{Z}}\mathbf{M}_1\mathbf{M}_3$ (возможны и другие преобразования) мы можем перейти от представления $\bar{\mathbf{Z}} = \mathbf{ZZZZZ}$ к представлению с весом три:

$$\begin{aligned}
\bar{\mathbf{Z}} &= \mathbf{ZZZZZ},\\
\mathbf{M}_1 &= \mathbf{XZZX1},\\
\mathbf{M}_3 &= \mathbf{X1XZZ},\\
\bar{\mathbf{Z}}' &= -\mathbf{Z1XX1}.
\end{aligned}$$

Следует быть внимательным при перемножении в четвертом столбце: правильный общий знак наверняка получится при разбиении умножения на два шага $\mathbf{ZX} = \mathbf{Y}$, $\mathbf{YZ} = -\mathbf{X}$.

b) Теоретик, занимающийся скрытыми переменными, хотел бы знать результат измерения $\mathbf{ZZZZZ}$ без его фактического выполнения. Скорее, он (или она) хотел бы *сделать вывод* о значении этой наблюдаемой, используя измерение некоторой из ее подсистем и знание некоторого глобального свойства состояния, то есть примерно в том же духе, как и в мысленном ЭПР-эксперименте, в начале которого известно, что две частицы имеют суммарный спин, равный нулю, а затем предпринимается попытка сделать вывод о значении $\boldsymbol{\sigma}_x^{(1)}$ (или $\boldsymbol{\sigma}_x^{(2)}$) по результату измерения $\boldsymbol{\sigma}_x^{(2)}$ (или $\boldsymbol{\sigma}_x^{(1)}$).

Рассмотрим систему, первоначально приготовленную как общее собственное пространство найденных в части **(a)** пяти циклически связанных

наблюдаемых

$$\begin{aligned}\bar{\mathbf{Z}}_0 &= -\mathbf{Z1XX1},\\ \bar{\mathbf{Z}}_1 &= -\mathbf{1Z1XX},\\ \bar{\mathbf{Z}}_2 &= -\mathbf{X1Z1X},\\ \bar{\mathbf{Z}}_3 &= -\mathbf{XX1Z1},\\ \bar{\mathbf{Z}}_4 &= -\mathbf{1XX1Z}.\end{aligned}$$

Теоретик, занимающийся скрытыми переменными, замечает, что о собственном значении $\mathbf{Z}$ на i-ом кубите можно сделать вывод, измеряя $\mathbf{X}$ на кубитах $(i+2) \bmod 5$ и $(i+3) \bmod 5$ и зная (глобальное) собственное число m_i наблюдаемой $\bar{\mathbf{Z}}_i$. Он (или она) доказывает, что, измеряя $\mathbf{XXXXX}$, можно сделать вывод о собственном значении $\mathbf{ZZZZZ}$, фактически не измеряя его. Следовательно, предсказание состоит в том, что собственное значение $\bar{z}$ наблюдаемой $\mathbf{ZZZZZ}$ связано с собственными значениями x_i наблюдаемой $\mathbf{X}_{(i)}$ и с m_i соотношением

$$\begin{aligned}\bar{z} &= \left(\frac{-m_0}{x_2x_3}\right)\left(\frac{-m_1}{x_3x_4}\right)\left(\frac{-m_2}{x_4x_0}\right)\left(\frac{-m_3}{x_0x_1}\right)\left(\frac{-m_4}{x_1x_2}\right) =\\ &= -(m_0m_1m_2m_3m_4)\left(x_0^2x_1^2x_2^2x_3^2x_4^2\right)^{-1} =\\ &= -(m_0m_1m_2m_3m_4).\end{aligned}$$

Для состояния $|\bar{0}\rangle$ $m_0 = m_1 = m_2 = m_3 = m_4 = +1$ так, что предсказывается $\bar{z} = -1$. Однако это находится в прямом противоречии с квантовомеханическим результатом, который предсказывает, что $\mathbf{ZZZZZ}|\bar{0}\rangle = +1 \cdot |\bar{0}\rangle$.

c) Эйнштейн сказал бы, что приведенное выше доказательство устанавливает экспериментально проверяемое различие между двумя эпистемиологическими точками зрения, поддерживаемыми теорией скрытых переменных и квантовой механикой соответственно. Он мог бы добавить, что его совместный с Розеном и Подольским первоначальный пример демонстрирует то же самое различие, но гораздо проще для понимания.

7.5. Обобщенный код Шора

a) Задача лучше решается на словах, чем в громоздкой записи. Концептуально, стабилизирующими операторами являются $m-1$ операторов $\mathbf{ZZ}$,

действующих на ближайшие соседние кубиты внутри каждого из m блоков, плюс $m-1$ операторов $\mathbf{XX}\ldots\mathbf{X}$, действующих на каждую пару ближайших соседних блоков. Закодированные операции представляют собой $\bar{\mathbf{X}}$, который инвертирует закодированный бит, обращая фазы каждого из блоков (используя один оператор $\mathbf{Z}$ на каждый блок), и $\bar{\mathbf{Z}}$, который обращает закодированную фазу, инвертируя все биты внутри блока (используя $\mathbf{X}^{\otimes m}$). В громоздких обозначениях мы можем использовать двойной индекс кубитов: первый индекс помечает, в каком блоке находится кубит, а второй — позицию кубита в этом блоке. Принимая эти обозначения, мы имеем

$$\begin{aligned}
\mathbf{M}_{i,j}^{(z)} &= \mathbf{Z}_{i,j}\mathbf{Z}_{i,j+1}, \quad i=1,\ldots,m, \quad j=1,\ldots,m-1,\\
\mathbf{M}_{i}^{(x)} &= \mathbf{X}_{i,1}\cdots\mathbf{X}_{i,m}\mathbf{X}_{i+1,1}\cdots\mathbf{X}_{i+1,m}, \quad i=1,\ldots,m-1,\\
\bar{\mathbf{X}} &= \mathbf{Z}_{1,1}\cdots\mathbf{Z}_{m,1},\\
\bar{\mathbf{Z}} &= \mathbf{X}_{1,1}\cdots\mathbf{X}_{1,m}.
\end{aligned}$$

Заметим, что, как и ожидалось, существует $m(m-1)+m-1=m^2-1=$ $=n-k$ генераторов стабилизатора.

b) Закодированный оператор минимального веса ($\bar{\mathbf{Z}}$) имеет вес m, так что расстояние кода равно m. Непосредственной проверкой можно убедиться в том, что не может быть других операций с более низким весом.

c) 1) Ошибка $\bar{\mathbf{X}}$ требует, чтобы более чем в половине блоков произошло обращение фазы (с помощью оператора $\mathbf{Z}$). Событие, которое выполняет это в главном порядке по p, состоит в том, что в отдельных блоках появляется *точно* $(m+1)/2$ ошибок $\mathbf{Z}$. Вероятность этого события равна

$$\begin{aligned}
P_x &= \left(\begin{array}{c} m \\ \frac{m+1}{2} \end{array}\right)^{\text{блоки}} \left(\begin{array}{c} m \\ 1 \end{array}\right)^{\text{кубиты/блоки}} {}^{(m+1)/2\ \text{блоки}}\\
&\quad \times \left(\frac{p}{3}[\text{для ошибки } \mathbf{Z}] + \frac{p}{3}[\text{для ошибки } \mathbf{X}]\right)^{(m+1)/2\ \text{ошибки}}\\
&= \left(\begin{array}{c} m \\ \frac{m+1}{2} \end{array}\right)\left(\begin{array}{c} m \\ 1 \end{array}\right)^{(m+1)/2}\left(\frac{2p}{3}\right)^{(m+1)/2}.
\end{aligned}$$

2) Ошибка $\bar{\mathbf{Z}}$ требует, чтобы более чем в половине в нечетном количестве блоков произошло инвертирование их битов (с помощью операторов $\mathbf{X}$). Событие, которое выполняет это в главном порядке по p, состоит в том, что в одном блоке появляется *точно* $(m+1)/2$ ошибок $\mathbf{X}$. Вероятность этого события равна

$$P_z = \underbrace{\binom{m}{1}}_{\text{блоки}} \underbrace{\binom{m}{\frac{m+1}{2}}}_{\text{кубиты/блоки}}^{\text{1 блок}} \times$$

$$\times \underbrace{\left(\frac{p}{3}[\text{для ошибки } \mathbf{Z}] + \frac{p}{3}[\text{для ошибки } \mathbf{X}]\right)^{(m+1)/2}}_{\text{ошибки}} =$$

$$= \binom{m}{1}\binom{m}{\frac{m+1}{2}}\left(\frac{2p}{3}\right)^{(m+1)/2}.$$

Заметим, что $P_x = m^{(m-1)/2} P_z$.

d) При больших m мы можем воспользоваться формулой Стирлинга

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n,$$

чтобы упростить выражения, найденные в части **(c)**. В этом приближении мы имеем

$$\binom{m}{\frac{m+1}{2}} \approx \binom{m}{m/2} =$$

$$= \frac{m!}{\frac{m}{2}!\frac{m}{2}!} \approx$$

$$\approx \frac{\sqrt{2\pi m}\left(\frac{m}{e}\right)^m}{2\pi\frac{m}{2}\left(\frac{m}{2e}\right)^{m/2}\left(\frac{m}{2e}\right)^{m/2}} =$$

$$= \sqrt{\frac{2}{\pi m}}2^m.$$

Следовательно, вероятность P_z приближенно равна

$$P_z \approx \sqrt{\frac{2m}{\pi}} 2^m \left(\frac{2p}{3}\right)^{(m+1)/2} =$$
$$= \sqrt{\frac{m}{2\pi}} 4^{(m+1)/2} \left(\frac{2p}{3}\right)^{(m+1)/2} =$$
$$= \sqrt{\frac{m}{2\pi}} \left(\frac{8p}{3}\right)^{(m+1)/2},$$

аналогично, вероятность P_x приближенно равна

$$P_x \approx \sqrt{\frac{m}{2\pi}} m^{(m-1)/2} \left(\frac{8p}{3}\right)^{(m+1)/2} =$$
$$= \sqrt{\frac{1}{2\pi m}} \left(\frac{8mp}{3}\right)^{(m+1)/2}.$$

Чтобы обеспечить хорошую защиту против ошибок обращения фазы при $m \to \infty$, нам необходимо, чтобы $p < 3/8$, так как тогда $P_z \to 0$. Чтобы обеспечить хорошую защиту против ошибок инвертирования бита при $m \to \infty$, нам необходимо, чтобы $p < 3/(8m)$, так как тогда $P_x \to 0$. Однако при $m \to \infty$ это требование эквивалентно требованию $p = 0$. Следовательно, это асимптотическое семейство кодов не может обеспечить надежную защиту против ошибок инвертирования бита, хотя может защитить от ошибок обращения фазы, если $p < 3/8$.

7.6. Кодирующие схемы

На лекциях обсуждение вопросов, касающихся кодирующих схем, было довольно кратким, поэтому здесь я его расширю. Прежде чем объяснить, как они работают, я представлю универсальный алгоритм конструирования кодирующей схемы данного стабилизирующего кода. Алгоритм естественным образом делится на три фазы.

Фаза I

1) Выразим стабилизатор на языке двоичного векторного пространства как $(H_x|H_z)$. Выполним процедуру исключения Гаусса–Жордана, так чтобы H_x начинался с единичной матрицы ранга r.

2) Для каждой закодированной операции $\bar{\mathbf{X}}_i$ найдем представление $\bar{\mathbf{X}}_i = \tilde{\mathbf{Z}}^{\otimes r}\tilde{\mathbf{X}}^{\otimes(n-k+r)}\mathbf{X}_{(n-k+i)}$, где $\tilde{\mathbf{Z}}$ равно $\mathbf{Z}$ или $\mathbf{1}$, а $\tilde{\mathbf{X}}$ равно $\mathbf{X}$ или $\mathbf{1}$. Такое представление всегда может быть найдено.[1] В литературе это представление известно как «стандартная форма» операторов $\bar{\mathbf{X}}_i$.

3) Проведем n горизонтальных контрольных линий схемы; начиная с верхней, снабдим их метками $1, \ldots, n$. Пусть k кодируемых кубитов занимают k нижних контрольных линий. Пусть $|0\rangle$ занимают остальные контрольные линии.

4) Изобразим каждый из операторов $\bar{\mathbf{X}}_i$ из пункта 2), последовательно действующих на кубиты, но заменим каждый сомножитель $\tilde{\mathbf{Z}}$ единицей, а $\mathbf{X}_{(n-k+i)}$ — контрольным узлом, управляющим остальной частью оператора $\bar{\mathbf{X}}_i$.

Фаза II

Изобразим поворот Адамара на каждой из первых r контрольных линий.

Фаза III

Изобразим первые r генераторов стабилизатора $\mathbf{M}_1, \ldots, \mathbf{M}_r$ [а именно, первые r строк матрицы $(H_x|H_z)$, приведенной с помощью выполненной в пункте 1 фазы I процедуры Гаусса–Жордана], последовательно действующих на кубиты, но в каждом $\mathbf{M}_i$ заменим множитель $\mathbf{X}_{(i)}$ контрольным узлом, управляющим остальной частью генератора $\mathbf{M}_i$. Более того, заменим в $\mathbf{M}_i$ на $\mathbf{1}$ каждый сомножитель $\mathbf{Z}_j$ с номером $j > i$. Если в $\mathbf{M}_i$ присутствует $\mathbf{Y}_{(i)}$, а не $\mathbf{X}_{(i)}$, то перед применением оператора контролируемое $\mathbf{M}_i$ предварительно умножим контрольный узел на оператор $\mathbf{Z}$.

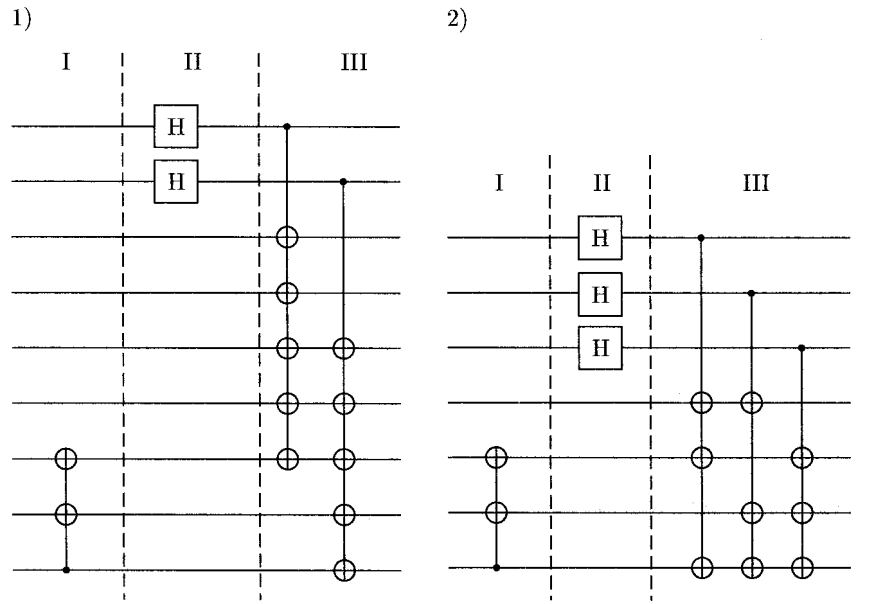
Фазы I и II всегда могут выполняться параллельно, несмотря на то, что концептуально они различны. Заметим, что фаза II полностью исчезает при $n - k = r$, как это имеет место в рассмотренном на лекции случае кода $[[5, 1, 3]]$. Однако этот этап нельзя забывать в случае кодирующих схем для кодов $[[9, 1, 3]]$ и $[[7, 1, 3]]$.

[1] Подробности можно найти в диссертации Д. Готтесмана *Stabilizier Codes and Quantum Error Correction*, quant-ph/9705052. Я не хочу здесь доказывать это утверждение.

Анализ алгоритма кодирования

Анализ того, почему это работает, увел бы нас далеко в сторону и сделал бы еще длиннее это уже и без того длинное решение. Интересующегося читателя я отсылаю к диссертации Даниэля Готтесмана *Stabilizier Codes and Quantum Error Correction*, доступной на домашней странице этого курса. Однако некоторые диаграммы в диссертации неправильны. В частности, в них не учитывается тот факт, что всякий раз, когда $j > i$, в $\mathbf{M}_i$ необходимо заменять $\mathbf{Z}_j$ на $\mathbf{1}$. Для предлагаемых здесь задач этот нюанс не важен, так как вся контролирующая логика осуществляется только с помощью операторов $\mathbf{X}$.

Применяя этот алгоритм к кодам Шора и Стина, мы генерируем кодировщики:



7.7. Укорачивание квантового кода

а) Допустим, что на последний кубит нетривиально действует более двух генераторов. Выберем один из них, назовем его для определенности $\mathbf{M}_1$ и выполним на последнем кубите локальное унитарное преобразование так, чтобы этот генератор действовал здесь как $\mathbf{X}$. Затем умножим на $\mathbf{M}_1$ каж-

дый другой генератор, действующий на последний кубит как **X** или **Y**. Если после этого шага не осталось других генераторов с нетривиальным носителем на последнем кубите, то мы выполнили свою работу. В противном случае, в соответствии с предыдущим действием $\mathbf{M}_1$, все такие оставшиеся генераторы должны действовать на последний кубит как **Z**. Выберем один из этих генераторов и назовем его $\mathbf{M}_2$. Умножим на $\mathbf{M}_2$ каждый другой генератор, действующий на последний кубит как **Z**. Получающийся в результате стабилизатор имеет самое большее два (а именно, $\mathbf{M}_1$ и $\mathbf{M}_2$) генератора, нетривиально действующих на последний кубит.

b) Это утверждение трудно доказать, поскольку оно неверно. В качестве контрпримера рассмотрим «выколотый»[1] код Шора $[[9,1,3]]$. Он не является кодом $[[8,2,2]]$, как это утверждается в условии. В действительности, он представляет собой код $[[8,2,1]]$, так как минимальный вес выкалываемой закодированной операции равен единице:

$$
\begin{array}{l}
\mathbf{M}_1 = \mathbf{Z}\ \mathbf{Z}\ 1\ 1\ 1\ 1\ 1\ 1 \\
\mathbf{M}_2 = 1\ \mathbf{Z}\ \mathbf{Z}\ 1\ 1\ 1\ 1\ 1\ 1 \\
\mathbf{M}_3 = 1\ 1\ 1\ \mathbf{Z}\ \mathbf{Z}\ 1\ 1\ 1\ 1 \\
\mathbf{M}_4 = 1\ 1\ 1\ 1\ \mathbf{Z}\ \mathbf{Z}\ 1\ 1\ 1 \\
\mathbf{M}_5 = 1\ 1\ 1\ 1\ 1\ 1\ \mathbf{Z}\ \mathbf{Z}\ 1 \\
\mathbf{M}_6 = 1\ 1\ 1\ 1\ 1\ 1\ 1\ \mathbf{Z}\ \mathbf{Z} \\
\mathbf{M}_7 = \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ 1\ 1\ 1 \\
\mathbf{M}_8 = 1\ 1\ 1\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X} \\
\bar{\mathbf{X}} = \mathbf{Z}\ 1\ 1\ \mathbf{Z}\ 1\ 1\ \mathbf{Z}\ 1\ 1 \\
\bar{\mathbf{Z}} = \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ 1\ 1\ 1\ 1\ 1\ 1
\end{array}
\quad \Rightarrow \quad
\begin{array}{l}
\mathbf{M}'_1 = \mathbf{Z}\ \mathbf{Z}\ 1\ 1\ 1\ 1\ 1\ 1 \\
\mathbf{M}'_2 = 1\ \mathbf{Z}\ \mathbf{Z}\ 1\ 1\ 1\ 1\ 1 \\
\mathbf{M}'_3 = 1\ 1\ 1\ \mathbf{Z}\ \mathbf{Z}\ 1\ 1\ 1 \\
\mathbf{M}'_4 = 1\ 1\ 1\ 1\ \mathbf{Z}\ \mathbf{Z}\ 1\ 1 \\
\mathbf{M}'_5 = 1\ 1\ 1\ 1\ 1\ 1\ \mathbf{Z}\ \mathbf{Z} \\
\mathbf{M}'_6 = \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ 1\ 1 \\
\bar{\mathbf{X}}'_1 = \mathbf{Z}\ 1\ 1\ \mathbf{Z}\ 1\ 1\ \mathbf{Z}\ 1 \\
\bar{\mathbf{Z}}'_1 = \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ 1\ 1\ 1\ 1\ 1 \\
\bar{\mathbf{X}}'_2 = 1\ 1\ 1\ 1\ 1\ 1\ 1\ \mathbf{Z} \\
\bar{\mathbf{Z}}'_2 = 1\ 1\ 1\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}
\end{array}
$$

Результат правилен, если ограничиться выкалыванием невырожденных кодов. Я докажу это от противного:

Утверждение: Выколотый код $[[n,k,d]]$ представляет собой код $[[n-1,k+1,d^*]]$, где $d^* \geqslant d-1$.

Доказательство: Пусть C является кодом $[[n,k,d]]$. Согласно результатам части **(a)** мы знаем, что выкалывание C дает в результате код C' $[[n-1,k+1,d^*]]$. Пусть d^* является расстоянием кода C'; предположим, что $d^* < d-1$. Покажем, что это предположение ведет к противоречию (и, следовательно, докажем утверждение).

Согласно предположению, существует оператор $\mathbf{E}'$ с весом d^*, коммутирующий со всеми генераторами стабилизатора кода C'. Оператор $\mathbf{E}'$

[1] Эта терминология позаимствована из теории классических кодов коррекции ошибок. Эту задачу лучше было озаглавить «Выкалывание квантового кода». Процесс укорачивания кода представляет собой другую операцию.

можно расширить до $\mathbf{E}$, который действует на C и имеет вес, не превышающий $d^* + 1$. Чтобы выполнить это, заметим сначала, что результаты части **(a)** говорят нам о том, что в самом общем виде генераторы C могут быть записаны как

$$\begin{aligned}\mathbf{M}_i &= \mathbf{M}'_i \otimes \mathbf{1},\\ \mathbf{M}_{n-k-1} &= \bar{\mathbf{X}}'_{k+1} \otimes \mathbf{X},\\ \mathbf{M}_{n-k} &= \bar{\mathbf{Z}}'_{k+1} \otimes \mathbf{Z},\end{aligned}$$

где $\mathbf{M}'_i$ $(i = 1, \ldots, n-k-2)$ — генераторы стабилизатора C', а $\bar{\mathbf{X}}'_{k+1}$ и $\bar{\mathbf{Z}}'_{k+1}$ — закодированные $\mathbf{X}$ и $\mathbf{Z}$ операторы, действующие на логический кубит $k+1$.

Построим $\mathbf{E}$, определяя его в зависимости от того, как оператор $\mathbf{E}' \otimes \mathbf{1}$ коммутирует с генераторами $\mathbf{M}_{n-k-1}$ и $\mathbf{M}_{n-k}$, следующим способом[1] (см. таблицу).

$[\mathbf{E}' \otimes \mathbf{1},\ \mathbf{M}_{n-k-1}]$	$[\mathbf{E}' \otimes \mathbf{1},\ \mathbf{M}_{n-k}]$	$\mathbf{E}$	Вес
$+1$	$+1$	$\mathbf{E}' \otimes \mathbf{1}$	d^*
$+1$	-1	$\mathbf{E}' \otimes \mathbf{X}$	$d^* + 1$
-1	$+1$	$\mathbf{E}' \otimes \mathbf{Z}$	$d^* + 1$
-1	-1	$\mathbf{E}' \otimes \mathbf{Y}$	$d^* + 1$

Эта конструкция гарантирует, что $\mathbf{E}$ коммутирует со всеми генераторами C' и имеет вес, меньший или равный $d^* + 1$. Но подождите! Так как C является кодом с расстоянием d, все ошибки с весом, не превосходящим $d-1$, или антикоммутируют с некоторым генератором стабилизатора C, или сами содержатся в C. Однако мы только что показали, что

$$\begin{aligned}\mathrm{wt}(\mathbf{E}) &\leqslant d^* - 1 <\\ &< d - 2 \not\geqslant\\ &\not\geqslant d - 1,\end{aligned}$$

и тем не менее $\mathbf{E}$ коммутирует со всеми генераторами стабилизатора C! Тогда $\mathbf{E}$ на самом деле *должен принадлежать* стабилизатору C, что имело бы место, если бы C был вырожденным. Но мы взяли C невырожденным, то есть все элементы его стабилизатора имеют веса, превосходящие d. Следовательно, полученное выше неравенство представляет искомое нами противоречие. □

c) Выкалывание кода $[[5, 1, 3]]$ дает код $[[4, 2, 2]]$, о чем говорилось на лекциях. Сначала мы перегруппируем элементы нормализатора кода $[[5, 1, 3]]$,

[1] Значения $+1(-1)$ в таблице означают, что операторы коммутируют (антикоммутируют).

при необходимости умножая генераторы на $\mathbf{M}_1$ или на $\mathbf{M}_2$,

$$\begin{array}{l} \mathbf{M}_1 = \mathbf{X}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{X}\ 1 \\ \mathbf{M}_2 = 1\ \mathbf{X}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{X} \\ \mathbf{M}_3 = \mathbf{X}\ 1\ \mathbf{X}\ \mathbf{Z}\ \mathbf{Z} \\ \mathbf{M}_4 = \mathbf{Z}\ \mathbf{X}\ 1\ \mathbf{X}\ \mathbf{Z} \\ \bar{\mathbf{X}} = \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X} \\ \bar{\mathbf{Z}} = \mathbf{Z}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{Z} \end{array} \quad \Rightarrow \quad \begin{array}{l} \mathbf{M}_1 = \ \mathbf{X}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{X}\ 1 \\ \mathbf{M}_2 = -\mathbf{Y}\ \mathbf{X}\ \mathbf{X}\ \mathbf{Y}\ 1 \\ \mathbf{M}_3 = \ 1\ \mathbf{X}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{X} \\ \mathbf{M}_4 = \ \mathbf{Z}\ \mathbf{X}\ 1\ \mathbf{X}\ \mathbf{Z} \\ \bar{\mathbf{X}} = \ \mathbf{X}\ 1\ \mathbf{Y}\ \mathbf{Y}\ 1 \\ \bar{\mathbf{Z}} = \ 1\ \mathbf{Y}\ \mathbf{Z}\ \mathbf{Y}\ 1 \end{array}$$

Затем мы обрезаем последний бит и для удобства устанавливаем общую для всех генераторов фазу $+1$ (результирующий код изоморфен исходному с точностью до произвольной общей фазы). Генераторы $\mathbf{M}_3$ и $\mathbf{M}_4$ превращаются в закодированные операторы $\bar{\mathbf{Z}}$ и $\bar{\mathbf{X}}$ для дополнительного закодированного кубита:

$$\begin{array}{l} \mathbf{M}_1 = \mathbf{X}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{X}\ 1 \\ \mathbf{M}_2 = \mathbf{Y}\ \mathbf{X}\ \mathbf{X}\ \mathbf{Y}\ 1 \\ \mathbf{M}_3 = 1\ \mathbf{X}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{X} \\ \mathbf{M}_4 = \mathbf{Z}\ \mathbf{X}\ 1\ \mathbf{X}\ \mathbf{Z} \\ \bar{\mathbf{X}} = \mathbf{X}\ 1\ \mathbf{Y}\ \mathbf{Y}\ 1 \\ \bar{\mathbf{Z}} = 1\ \mathbf{Y}\ \mathbf{Z}\ \mathbf{Y}\ 1 \end{array} \quad \text{выкалывание} \quad \longrightarrow \quad \begin{array}{l} \mathbf{M}_1 = \mathbf{X}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{X} \\ \mathbf{M}_2 = \mathbf{Y}\ \mathbf{X}\ \mathbf{X}\ \mathbf{Y} \\ \bar{\mathbf{X}}_1 = 1\ \mathbf{X}\ \mathbf{Z}\ \mathbf{Z} \\ \bar{\mathbf{Z}}_1 = \mathbf{Z}\ \mathbf{X}\ 1\ \mathbf{X} \\ \bar{\mathbf{X}}_2 = \mathbf{X}\ 1\ \mathbf{Y}\ \mathbf{Y} \\ \bar{\mathbf{Z}}_2 = 1\ \mathbf{Y}\ \mathbf{Z}\ \mathbf{Y} \end{array}$$

Мы вправе изменить базисы некоторых кубитов, то есть применить одновременные ЛУПы $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y} \to \mathbf{X}$ к первому и последнему кубитам, чтобы получить эквивалентный стабилизатор:

$$\begin{array}{l} \mathbf{M}_1 = \mathbf{X}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{X} \\ \mathbf{M}_2 = \mathbf{Y}\ \mathbf{X}\ \mathbf{X}\ \mathbf{Y} \\ \bar{\mathbf{X}}_1 = 1\ \mathbf{X}\ \mathbf{Z}\ \mathbf{Z} \\ \bar{\mathbf{Z}}_1 = \mathbf{Z}\ \mathbf{X}\ 1\ \mathbf{X} \\ \bar{\mathbf{X}}_2 = \mathbf{X}\ 1\ \mathbf{Y}\ \mathbf{Y} \\ \bar{\mathbf{Z}}_2 = 1\ \mathbf{Y}\ \mathbf{Z}\ \mathbf{Y} \end{array} \quad \text{ЛУПы} \quad \Rightarrow \quad \begin{array}{l} \mathbf{M}_1 = \mathbf{Z}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{Z} \\ \mathbf{M}_2 = \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X} \\ \bar{\mathbf{X}}_1 = 1\ \mathbf{X}\ \mathbf{Z}\ \mathbf{Y} \\ \bar{\mathbf{Z}}_1 = \mathbf{Y}\ \mathbf{X}\ 1\ \mathbf{Z} \\ \bar{\mathbf{X}}_2 = \mathbf{Z}\ 1\ \mathbf{Y}\ \mathbf{X} \\ \bar{\mathbf{Z}}_2 = 1\ \mathbf{Y}\ \mathbf{Z}\ \mathbf{X} \end{array}$$

Заменяя закодированные операторы их двойниками с весом два, мы можем сделать более очевидным, что расстояние проколотого кода равно двум. Одним из способов выполнения этого является замена:

$$\begin{array}{lll} \bar{\mathbf{X}}_1 & \to & \bar{\mathbf{X}}_1\bar{\mathbf{Z}}_2\bar{\mathbf{X}}_2\mathbf{M}_1 \quad = -11\mathbf{XX} \\ \bar{\mathbf{Z}}_1 & \to & \bar{\mathbf{X}}_1\bar{\mathbf{Z}}_2 \quad = -1\mathbf{Z}1\mathbf{Z} \\ \bar{\mathbf{X}}_2 & \to & \bar{\mathbf{X}}_2\bar{\mathbf{Z}}_1\bar{\mathbf{X}}_1 \quad = -\mathbf{X}1\mathbf{X}1 \\ \bar{\mathbf{Z}}_2 & \to & \bar{\mathbf{X}}_2\bar{\mathbf{Z}}_1\mathbf{M}_2 \quad = -11\mathbf{ZZ} \end{array}$$

Заметим, что эти преобразования сохраняют коммутационные соотношения между логическими операциями. В качестве заключительного шага мы

можем удалить общие фазы в нормализаторе:

$$\begin{aligned}
\mathbf{M}_1 &= \mathbf{Z}\ \mathbf{Z}\ \mathbf{Z}\ \mathbf{Z} \\
\mathbf{M}_2 &= \mathbf{X}\ \mathbf{X}\ \mathbf{X}\ \mathbf{X} \\
\bar{\mathbf{X}}_1 &= \mathbf{1}\ \mathbf{1}\ \mathbf{X}\ \mathbf{X} \\
\bar{\mathbf{Z}}_1 &= \mathbf{1}\ \mathbf{Z}\ \mathbf{1}\ \mathbf{Z} \\
\bar{\mathbf{X}}_2 &= \mathbf{X}\ \mathbf{1}\ \mathbf{X}\ \mathbf{1} \\
\bar{\mathbf{Z}}_2 &= \mathbf{1}\ \mathbf{1}\ \mathbf{Z}\ \mathbf{Z}
\end{aligned}$$

Используя это представление закодированных операций, чтобы выбрать базис для кодовых слов, мы находим, что кодовыми словами кода $[[4,2,2]]$ являются

$$|\bar{0}\bar{0}\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle),$$

$$|\bar{0}\bar{1}\rangle = \frac{1}{\sqrt{2}}(|1010\rangle + |0101\rangle),$$

$$|\bar{1}\bar{0}\rangle = \frac{1}{\sqrt{2}}(|0011\rangle + |1100\rangle),$$

$$|\bar{1}\bar{1}\rangle = \frac{1}{\sqrt{2}}(|1001\rangle + |0110\rangle).$$

Может быть, непосредственно этот код и не знаком, но его структура — да. Каждое кодовое слово является суперпозицией строк четного веса и длины четыре. Из классического кодирования Рида–Маллера нам известно, что $R(m-1,m)$ является пространством всех строк четного веса и длины 2^m. Следовательно, вышеприведенные квантовые кодовые слова выглядят, как смежные классы пространства $R(1,2)$ и другого, входящего в него, кода. Это подозрение подтверждается, если мы применим конструкцию КШС к $R(1,2)$ и к дуальному к нему $R(0,2)$. (Вспомним, что $R^{\perp}(r,m) = R(m-r-1,m)$.) Поскольку $R(0,2) \subseteq R(1,2)$ [так как $R(r,m)$ представляет полиномы степени r над $\mathbb{F}_m$], конструкция КШС справедлива и дает квантовый код с параметрами $[[4,2,2]]$. Похоже, что этот код хорошо описывается как «квантовый код Рида–Маллера».

7.8. Коды для кудитов

а)

1) Да, $\{\mathbf{E}_{r,s}\}$ образуют базис для $\mathrm{U}(d)$. Чтобы доказать это, мы должны показать, что они являются линейной оболочкой $\mathrm{U}(d)$. Достаточно показать, что базис $\{|a\rangle\langle b|\}$ для $\mathrm{U}(d)$ может быть разложен по $\{\mathbf{E}_{r,s}\}$, так как оба множества операторов имеют размер $d^2 = \dim \mathrm{U}(d)$.

Заметим сначала, что $\mathbf{E}_{r,s}$ может быть записан в базисе $\{|a\rangle\langle b|\}$ как[1]

$$\mathbf{E}_{r,s} = \sum_{j=0}^{d-1} \omega^{js} |j+r\rangle\langle j|.$$

Базисный элемент $|a\rangle\langle b|$ имеет разложение

$$|a\rangle\langle b| = \sum_{r,s=0}^{d-1} c_{rs}\mathbf{E}_{r,s},$$

коэффициенты которого даются выражением

$$\begin{aligned} c_{rs} &= \frac{1}{d}\,\mathrm{tr}\left(\mathbf{E}_{r,s}^{\dagger}|a\rangle\langle b|\right) = \\ &= \frac{1}{d}\,\mathrm{tr}\left(\sum_{j=0}^{d-1}\omega^{-js}|j\rangle\langle j+r|a\rangle\langle b|\right) = \\ &= \frac{1}{d}\sum_{i,j=0}^{d-1}\omega^{-js}\langle i|j\rangle\langle j+r|a\rangle\langle b|i\rangle = \\ &= \frac{1}{d}\omega^{-bs}\delta_{a,b+r}. \end{aligned}$$

Чтобы убедиться в том, что это «разложение Фурье» правильно, заметим, что

$$\begin{aligned} \left\langle b\left|\sum_{r,s=0}^{d-1} c_{rs}\mathbf{E}_{r,s}\right|a\right\rangle &= \frac{1}{d}\sum_{r,s,j=0}^{d-1}\omega^{-bs+js}\delta_{a,b+r}\langle b|j+r\rangle\langle j|a\rangle = \\ &= \frac{1}{d}\sum_{r,s=0}^{d-1}\omega^{(a-b)s}\delta_{a,b+r}\delta_{b,a+r} = \\ &= \frac{1}{d}\sum_{r,s=0}^{d-1}\omega^{(a-b)s}\delta_{r,a-b}\delta_{r,b-a} = \\ &= \frac{1}{d}\sum_{s=0}^{d-1}(1)^s = \\ &= 1. \end{aligned}$$

[1] С этого момента все дополнительные сложения и вычитания в решениях интерпретируются по модулю d, где d — размерность рассматриваемого кудита.

2) Да, операторы $\{\mathbf{E}_{r,s}\}$ унитарны. Так как $\mathbf{X}$ и $\mathbf{Z}$ сохраняют внутреннее произведение, то таким же свойством обладает $\mathbf{E}_{r,s} = \mathbf{X}^r\mathbf{Z}^s$:

$$\begin{aligned}\langle i|\mathbf{X}^\dagger\mathbf{X}|j\rangle &= \langle i+1|j+1\rangle = \\ &= \delta_{ij}, \\ \langle i|\mathbf{Z}^\dagger\mathbf{Z}|j\rangle &= \langle i|\omega^{-is}\omega^{js}|j\rangle = \\ &= \omega^{(j-i)s}\langle i|j\rangle = \\ &= \delta_{ij}.\end{aligned}$$

3) След внутреннего произведения базисных элементов (которые явно использовались в пункте 1 этой задачи) равен

$$\begin{aligned}\mathrm{tr}(\mathbf{E}_{r,s}^\dagger\mathbf{E}_{t,u}) &= \mathrm{tr}(\mathbf{Z}^{-s}\mathbf{X}^{-r}\mathbf{X}^t\mathbf{Z}^u) = \\ &= \mathrm{tr}(\mathbf{X}^{t-r}\mathbf{Z}^{u-s}) = \\ &= \sum_{j=0}^{d-1}\langle j|\mathbf{X}^{t-r}\mathbf{Z}^{u-s}|j\rangle = \\ &= \sum_{j=0}^{d-1}\omega^{(u-s)j}\langle j|j+t-r\rangle = \\ &= \delta_{tr}\sum_{j=0}^{d-1}\omega^{(u-s)j} = \\ &= \begin{cases}\delta_{tr}\cdot d, & u=s, \\ \delta_{tr}\dfrac{1-\omega^{(u-s)d}}{1-\omega^{u-s}} = 0, & u\neq s\end{cases} = \\ &= d\cdot\delta_{tr}\delta_{us}.\end{aligned}$$

b) Вычисляя действие коммутатора[1] $[\mathbf{E}_{r,s},\mathbf{E}_{t,u}] = \mathbf{E}_{r,s}\mathbf{E}_{t,u}\mathbf{E}_{r,s}^{-1}\mathbf{E}_{t,u}^{-1}$ на пробное состояние $|j\rangle$, мы находим

$$\begin{aligned}[\mathbf{E}_{r,s},\mathbf{E}_{t,u}]|j\rangle &= \mathbf{X}^r\mathbf{Z}^s\mathbf{X}^t\mathbf{Z}^u\mathbf{Z}^{-s}\mathbf{X}^{-r}\mathbf{Z}^{-u}\mathbf{X}^{-t}|j\rangle = \\ &= \omega^{-u(j-t)-s(j-t-r)+u(j-t-r)+s(j-r)}|j\rangle = \\ &= \omega^{st-ur}|j\rangle.\end{aligned}$$

[1] Это алгебраическое определение коммутатора. Когда вычисляются синдромы, мы интересуемся именно *этим*, а не групповым коммутатором $[a,b] = ab - ba$.

Следовательно, $\eta_{r,s;t,u} = \omega^{st-ur} = \omega^{(t,u)*(r,s)}$, где $*$ обозначает определенное на лекциях симплектическое внутреннее произведение.

c) Всякий раз, когда мы добавляем генератор к стабилизирующему коду для кудитов, мы сокращаем размерность его кодового подпространства на множитель, равный порядку генератора. Если d не простое число, то возможно иметь генераторы порядка d_i, являющегося делителем d, но не самим d. После $n-k$ выборов генераторов размерность кодового подпространства становится

$$\begin{aligned} D &= \frac{d^n}{\prod_{i=1}^{n-k} d_i} = \\ &= \frac{d^n}{d^{n-k} \prod_{i=1}^{n-k} \frac{1}{r_i}} = \\ &= d^k \left(\prod_{i=1}^{n-k} r_i \right) \geqslant \\ &\geqslant d^k. \end{aligned}$$

Следовательно, такие коды могут кодировать более k кудитов! Это поднимает вопрос: что же тогда здесь закодировано? Чтобы ответить на него, исследуем два кода $d = 4$. Первый код, который мы изучим, имеет $n = 1$. Его стабилизатором является

$$\mathbf{M}_1 = \mathbf{X}^2$$

размерность кодового подпространства равна $d^n/|\mathbf{M}_1| = 4^1/2 = 2$. Этот код недостаточно велик, чтобы сохранять даже один кудит. Но он достаточно велик, чтобы вместить один кубит. Нормализатор кода содержит $\mathbf{X}$ и $\mathbf{Z}^2$, которые, мы подозреваем, имеют коммутационное соотношение

$$\begin{aligned} [\mathbf{X}, \mathbf{Z}^2] = \mathbf{X}\mathbf{Z}^2\mathbf{X}^{-1}\mathbf{Z}^{-2} &= \\ = \omega^{-2} &= \\ = -1. & \end{aligned}$$

Наше подозрение подтверждается — этот код $[[1,0,1]]_4$ кодирует один кубит:

$$|\bar{0}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |2\rangle),$$

$$|\bar{1}\rangle = \frac{1}{\sqrt{2}}(|1\rangle + |3\rangle).$$

Теперь более интересный пример. Рассмотрим код $[[5,1,3]]_4$ из задачи **7.11**. Квадраты всех генераторов образуют новый код со стабилизатором

$$\begin{array}{ccccc} \mathbf{X}^2 & \mathbf{Z}^2 & \mathbf{Z}^{-2} & \mathbf{X}^{-2} & \mathbf{1} \\ \mathbf{1} & \mathbf{X}^2 & \mathbf{Z}^2 & \mathbf{Z}^{-2} & \mathbf{X}^{-2} \\ \mathbf{X}^{-2} & \mathbf{1} & \mathbf{X}^2 & \mathbf{Z}^2 & \mathbf{Z}^{-2} \\ \mathbf{Z}^{-2} & \mathbf{X}^{-2} & \mathbf{1} & \mathbf{X}^2 & \mathbf{Z}^2 \end{array}$$

Размерность кодового подпространства равна $d^n / \prod_i |\mathbf{M}_i| = 4^5/2^4 = 4^3 = 64$. Он достаточно велик, чтобы сохранять три «кукварта», или два кукварта и два кубита, или один кукварт и четыре кубита и так далее. Какое решение принять? Оказывается, что мы добавляем к коду по одному кубиту для каждого генератора возведенного в квадрат исходного кода куквартов. Таким образом, в первом примере, где стабилизатором был просто $\mathbf{X}$, мы имели $k = 0$ куквартов. Тогда возведение в квадрат этого генератора привело к одному закодированному кубиту. В этом примере после возведения в квадрат четырех генераторов мы имеем $k = 1$ кукварт плюс четыре кубита. Доказательство этого общего факта увело бы нас даже еще дальше в сторону, но достаточно сказать, что для этого примера закодированными операциями являются:

один кудит:

$$\bar{\mathbf{X}}_4 = \mathbf{XXXXX},$$
$$\bar{\mathbf{Z}}_4 = \mathbf{ZZZZZ},$$

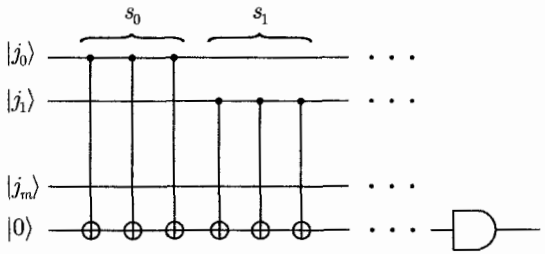
четыре кубита:

$$\bar{\mathbf{X}}_2^{(i)} = \mathbf{Z1XX1} \quad + \text{три перестановки},$$
$$\bar{\mathbf{Z}}_2^{(i)} = \mathbf{Z11ZX} \quad + \text{три перестановки}.$$

7.9. Измерение синдрома для кудита

а) Измерение наблюдаемой $\bigotimes_a \mathbf{Z}_a^{s_a}$ на $\bigotimes_a |j_a\rangle$ дает собственное значение $\omega^{\sum_a s_a j_a}$. Однако измерение $\mathbf{Z}$ на $|\sum_a s_a j_a\rangle$ также дает собственное значение $\omega^{\sum_a s_a j_a}$. Это подсказывает, что для того чтобы измерить $\bigotimes_a \mathbf{Z}_a^{s_a}$, мы должны применить s_a копий схемы SUM между каждым кубитом $|j_a\rangle$ и служебным кубитом, первоначально приготовленным в состоянии $|0\rangle$. Тогда мы можем измерить $\mathbf{Z}$ на служебном кубите, чтобы получить собствен-

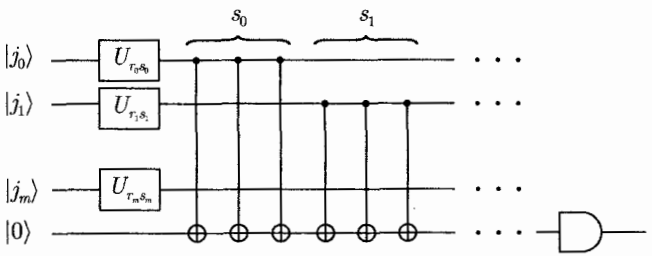
ное значение наблюдаемой $\bigotimes_a \mathbf{Z}_a^{s_a}$. Соответствующая схема имеет вид



b) Измерение наблюдаемой $\bigotimes_a \mathbf{E}_{r_a,s_a}$ на $\bigotimes_a |j_a\rangle$ дает такое же собственное значение, какое дает измерение $\bigotimes_a \mathbf{U}_{r_a,s_a} \mathbf{E}_{r_a,s_a} \mathbf{U}^\dagger_{r_a,s_a}$ на $\bigotimes_a \mathbf{U}_{r_a,s_a} |j_a\rangle$. Но поскольку (только для простых d!)

$$\bigotimes_a \mathbf{U}_{r_a,s_a} \mathbf{E}_{r_a,s_a} \mathbf{U}^\dagger_{r_a,s_a} = \mathbf{Z},$$

то все, что нам нужно сделать, — это модифицировать схему из части **(a)**, предварительно умножая каждый кубит $|j_a\rangle$ на $\mathbf{U}_{r_a,s_a}$:



7.10. Коды детектирования ошибок в кудитах

a) Да, генераторы коммутируют

$$\begin{aligned}[\mathbf{ZZZ}, \mathbf{XXX}] &= [\mathbf{Z}, \mathbf{X}]^3 = \\ &= \eta_{0,1;1,0}^3 = \\ &= 1.\end{aligned}$$

b) Расстояние этого кода равно двум. Оно равно минимальному весу приведенных ниже операторов нормализатора

$$\begin{aligned}\bar{\mathbf{X}} &= \mathbf{1}\ \mathbf{X}\ \mathbf{X}^2, \\ \bar{\mathbf{Z}} &= \mathbf{Z}^2\ \mathbf{Z}\ \mathbf{1}.\end{aligned}$$

Вместо того, чтобы доказывать, что эти представления имеют минимальный вес, равный двум, проще заметить, что расстояние кода должно быть больше единицы, так как все операторы с единичным весом антикоммутируют (то есть имеют нетривиальный синдром) по меньшей мере с одним из генераторов стабилизатора.

Достаточно составить список представленных выше операторов $\bar{\mathbf{X}}$ и $\bar{\mathbf{Z}}$, чтобы точно определить полный нормализатор, поскольку любой другой логический оператор $\bar{\mathbf{E}}_{r,s}$ может быть записан как $\bar{\mathbf{X}}^r\bar{\mathbf{Z}}^s$.

c) Ортонормированный базис для кодового подпространства можно получить, формируя сначала $|\bar{0}\rangle$, а затем необходимое количество раз применяя логические повышающие операторы $\bar{\mathbf{X}}$. Чтобы образовать $|\bar{0}\rangle$, обычно действуют на $|000\rangle$ суммой всех элементов стабилизатора. Но, поскольку для этого кода $\mathbf{ZZZ}$ действует на $|000\rangle$ как единица, нам на самом деле необходима только сумма по элементам стабилизатора, порождаемым оператором $\mathbf{XXX}$. Выполняя это, мы находим логические закодированные состояния

$$|\bar{0}\rangle = \frac{1}{\sqrt{3}}(|000\rangle + |111\rangle + |222\rangle),$$

$$|\bar{1}\rangle = \frac{1}{\sqrt{3}}(|012\rangle + |120\rangle + |201\rangle),$$

$$|\bar{2}\rangle = \frac{1}{\sqrt{3}}(|021\rangle + |102\rangle + |210\rangle).$$

Для быстрого запоминания заметим, что $|\bar{1}\rangle$ и $|\bar{2}\rangle$ представляют собой суперпозиции четных и нечетных перестановок в S_3 соответственно.

d) Мы можем обобщить предыдущие результаты, чтобы создать квантовый код $[[3m, 3m-2, 2]]_3$. Его стабилизатор генерируется операторами

$$\mathbf{M}_1 = \mathbf{X}^{\otimes 3m},$$
$$\mathbf{M}_2 = \mathbf{Z}^{\otimes 3m}.$$

Они коммутируют между собой, поскольку $\omega^{3m} = 1$. Я благодарен Дэвиду Бэкману, нашедшему нормализатор этого кода:

$$\begin{aligned}
\bar{\mathbf{X}}_1 &= 1\mathbf{X}\mathbf{X}^{-1} & \bar{\mathbf{Z}}_1 &= \mathbf{Z}^{-1}\mathbf{Z} \\
\bar{\mathbf{X}}_2 &= 11\mathbf{X}\mathbf{X}^{-1} & \bar{\mathbf{Z}}_2 &= \mathbf{Z}^{-2}\mathbf{Z}\mathbf{Z} \\
\bar{\mathbf{X}}_3 &= 111\mathbf{X}\mathbf{X}^{-1} & \bar{\mathbf{Z}}_3 &= \mathbf{Z}^{-3}\mathbf{Z}\mathbf{Z}\mathbf{Z} \\
&\vdots & &\vdots \\
\bar{\mathbf{X}}_{3m-2} &= 1^{\otimes(3m-2)}\mathbf{X}\mathbf{X}^{-1} & \bar{\mathbf{Z}}_{3m-2} &= \mathbf{Z}^{-(3m-2)}\mathbf{Z}^{\otimes(3m-2)}
\end{aligned}$$

е) Мы можем обобщить эти результаты еще дальше, рассматривая кудиты вместо кутритов. Стабилизатор для квантового кода $[[d, d-2, 2]]_d$ представляет собой

$$\begin{aligned}\mathbf{M}_1 &= \mathbf{X}^{\otimes d},\\ \mathbf{M}_2 &= \mathbf{Z}^{\otimes d},\end{aligned}$$

а его нормализатор генерируется операторами

$$\begin{aligned}
\bar{\mathbf{X}}_1 &= 1\mathbf{X}\mathbf{X}^{-1} & \bar{\mathbf{Z}}_1 &= \mathbf{Z}^{-1}\mathbf{Z}\\
\bar{\mathbf{X}}_2 &= 11\mathbf{X}\mathbf{X}^{-1} & \bar{\mathbf{Z}}_2 &= \mathbf{Z}^{-2}\mathbf{Z}\mathbf{Z}\\
\bar{\mathbf{X}}_3 &= 111\mathbf{X}\mathbf{X}^{-1} & \bar{\mathbf{Z}}_3 &= \mathbf{Z}^{-3}\mathbf{Z}\mathbf{Z}\mathbf{Z}\\
&\vdots & &\vdots\\
\bar{\mathbf{X}}_{d-2} &= 1^{\otimes(d-2)}\mathbf{X}\mathbf{X}^{-1} & \bar{\mathbf{Z}}_{d-2} &= \mathbf{Z}^{-(d-2)}\mathbf{Z}^{\otimes(d-2)}
\end{aligned}$$

7.11. Коды коррекции ошибок в кудитах

а) Порядок каждого генератора кода $[[5, 1, 3]]_d$ равен d. Это следует из того, что для каждого $\mathbf{E}_{r,s} \in \{\mathbf{X}, \mathbf{X}^{-1}, \mathbf{Z}, \mathbf{Z}^{-1}\}$ r и s являются взаимно простыми с d. Следовательно, каждый генератор $\mathbf{M}_i$ не может иметь порядок, который является делителем d, но не равен d. В качестве интересного добавления заметим, что поскольку *только* 1 и $d-1$ при всех d являются взаимно простыми с d, каждый генератор кода, который справедлив при любом d, в своем разложении на тензорные произведения должен содержать оператор из следующего множества:

$$\{1, \mathbf{X}, \mathbf{Z}, \mathbf{X}^{-1}, \mathbf{Z}^{-1}, \mathbf{X}\mathbf{Z}, \mathbf{X}\mathbf{Z}^{-1}, \mathbf{X}^{-1}\mathbf{Z}, \mathbf{X}^{-1}\mathbf{Z}^{-1}\}.$$

Все генераторы действительно независимы. Приравнивая произведения произвольных степеней генераторов единице, мы вынуждены положить равными нулю все показатели степеней (это доказательство является мультипликативным двойником обычного доказательства линейной независимости):

$$\begin{aligned}
1 &= \mathbf{M}_1^{c_1}\mathbf{M}_2^{c_2}\mathbf{M}_3^{c_3}\mathbf{M}_4^{c_4} =\\
&= (\mathbf{X}^{c_1-c_2}\mathbf{Z}^{-c_4}) \otimes (\mathbf{Z}^{c_1}\mathbf{X}^{c_2-c_4}) \otimes (\mathbf{Z}^{c_2-c_1}\mathbf{X}^{c_3}) \otimes\\
&\quad \otimes(\mathbf{X}^{-c_1}\mathbf{Z}^{c_3-c_2}\mathbf{X}^{c_4}) \otimes (\mathbf{X}^{c_2}\mathbf{Z}^{c_2-c_1}),\\
&\Rightarrow c_1 = c_2 = c_3 = c_4 = 0.
\end{aligned}$$

Все генераторы также коммутируют, в чем можно убедиться непосредственной проверкой: всякий раз, когда элемент $\mathbf{X}^a\mathbf{Z}^b$ выстраивается между двумя генераторами, еще один элемент $\mathbf{Z}^{-b}\mathbf{X}^{-a}$ также выстраивается таким образом, что в общем генераторы будут коммутировать.

Пятая циклическая перестановка *не является* независимой от остальных

$$\begin{aligned}\mathbf{Z}\mathbf{Z}^{-1}\mathbf{X}^{-1}\mathbf{1}\mathbf{X} &= (\mathbf{M}_1\mathbf{M}_2\mathbf{M}_3\mathbf{M}_4)^{-1} = \\ &= \mathbf{M}_4^{-1}\mathbf{M}_3^{-1}\mathbf{M}_2^{-1}\mathbf{M}_1^{-1} = \\ &= \mathbf{M}_1^{-1}\mathbf{M}_2^{-1}\mathbf{M}_3^{-1}\mathbf{M}_4^{-1}.\end{aligned}$$

b) Код $[[5,1,3]]_d$ невырожден. Чтобы понять это, заметим сначала, что его расстояние не больше трех, так как закодированные операции в части **c)** имеют вес три. Затем рассмотрим общий оператор Паули с весом $\leqslant 2$. Вследствие циклической природы кода этот оператор можно записать как $\mathbf{E}_a = \mathbf{E}_{r,s} \otimes \mathbf{E}_{t,u} \otimes \mathbf{1} \otimes \mathbf{1} \otimes \mathbf{1}$ или $\mathbf{E}_b = \mathbf{E}_{r,s} \otimes \mathbf{1} \otimes \mathbf{E}_{t,u} \otimes \mathbf{1} \otimes \mathbf{1}$.

В первом случае, чтобы коммутировать со стабилизатором, необходимо

$$\begin{aligned}[\mathbf{M}_1, \mathbf{E}_a] &= \omega^{-s+t} = 1, \\ [\mathbf{M}_2, \mathbf{E}_a] &= \omega^{u} = 1, \\ [\mathbf{M}_3, \mathbf{E}_a] &= \omega^{r} = 1, \\ [\mathbf{M}_4, \mathbf{E}_a] &= \omega^{-r+u} = 1, \\ &\Rightarrow r = s = t = u = 0.\end{aligned}$$

Во втором случае, чтобы коммутировать со стабилизатором, необходимо

$$\begin{aligned}[\mathbf{M}_1, \mathbf{E}_b] &= \omega^{-s-t} = 1, \\ [\mathbf{M}_2, \mathbf{E}_b] &= \omega^{t} = 1, \\ [\mathbf{M}_3, \mathbf{E}_b] &= \omega^{r-u} = 1, \\ [\mathbf{M}_4, \mathbf{E}_b] &= \omega^{-r} = 1, \\ &\Rightarrow r = s = t = u = 0.\end{aligned}$$

В обоих случаях это возможно только для единичного оператора. Следовательно, все операторы Паули с весом единица и два «антикоммутируют» с некоторым элементом стабилизатора и, следовательно, расстояние этого кода равно трем. Более того, это показывает, что не существует элемента стабилизатора, который может иметь вес меньше трех (так как он не может

коммутировать со всеми другими элементами стабилизатора), то есть код также является невырожденным.

c) Одним возможным представлением закодированных операций является

$$\bar{\mathbf{X}} = \mathbf{Z}^{-1}\,\mathbf{X}^{-1}\,\mathbf{Z}^{-1}\;\mathbf{1}\;\mathbf{1},$$
$$\bar{\mathbf{Z}} = \mathbf{X}\;\mathbf{X}\;\mathbf{1}\;\mathbf{Z}^{-1}\;\mathbf{1}.$$

Можно непосредственно проверить, что они имеют правильные коммутационные соотношения и коммутируют со всеми генераторами стабилизатора. Так как все сомножители включают только $\mathbf{Z}$, $\mathbf{X}$ и обратные к ним операторы, эти закодированные операции справедливы при любом d.

Глава 8

Топологические квантовые вычисления

8.1. Анионы?

Одной из главных идей квантовой теории является концепция *неразличимости частиц* (часто называемых *тождественными частицами*). Например, все электроны во Вселенной в точности одинаковы. Следовательно, для многоэлектронной системы операция *перестановки* двух электронов (обмена их положениями) является симметрией — она оставляет неизменной физику. Эту симметрию представляет унитарное преобразование, действующее на многоэлектронную волновую функцию.

Для неразличимых частиц в трехмерном пространстве, о котором мы обычно говорим в физике, перестановка частиц представляется одним из двух различных способов. Если частицы являются бозонами (например, атомы ^{4}He в сверхтекучей жидкости), то перестановку двух частиц представляет тождественный оператор: волновая функция инвариантна, то есть частицы подчиняются статистике Бозе. Если частицы являются фермионами (как, например, электроны в металлах), то перестановка представляется умножением на (-1): волновая функция меняет знак, то есть частицы подчиняются статистике Ферми.

В одномерном пространстве концепция статистики тождественных частиц становится неоднозначной. Дело в том, что в этом случае, обмениваясь местами, две частицы должны пройти друг сквозь друга. Если при перестановке двух тождественных частиц волновая функция меняет знак, то можно сказать, что они являются невзаимодействующими фермионами, но с тем же успехом можно сказать, что эти частицы являются взаимодействующими бозонами, так что изменение знака обусловлено взаимодействием проходящих друг сквозь друга частиц. В общем случае перестановка может изменить волновую функцию на мультипликативную фазу $e^{i\theta}$, принимающую значения, отличные от $+1$ и -1. Но даже такое изменение фазы можно учесть, описывая частицы или как бозоны, или как фермионы.

Таким образом, в пространстве трех (и большего числа) измерений, а также в одном измерении, статистика тождественных частиц довольно

проста. Но между этими двумя скучными случаями, в двумерном пространстве, возможно замечательно богатое разнообразие типов статистик частиц, настолько богатое, что мы просто утонем в нем, прежде чем сможем дать полезную классификацию всех возможностей.

Неразличимые частицы в двумерии, не являющиеся ни бозонами, ни фермионами, называются *анионами*. Анионы — это удивительно красивая теоретическая конструкция, но имеют ли они хоть какое-нибудь отношение к физике реальных, изучаемых в лаборатории, систем? Ответ замечателен: «Да!» Даже в нашем трехмерном мире можно создать двумерный электронный газ, удерживая электроны в тонком слое между двумя полупроводниковыми пластинами, так что при низких энергиях их движение в перпендикулярном слою направлении будет заморожено. Если электроны в материале достаточно подвижны, то в достаточно сильном магнитном поле и при достаточно низкой температуре такой двумерный электронный газ переходит в исключительно запутанное основное состояние, отделенное от всех возбужденных состояний отличной от нуля энергетической щелью. Более того, низкоэнергетические возбуждения частиц в таких системах не описываются электронными квантовыми числами; скорее, они являются анионами, несущими электрический заряд, равный дробной части заряда электрона. Анионы оказывают впечатляющее влияние на транспортные свойства образца, проявляющееся как дробный квантовый эффект Холла (ДКЭХ).

Анионы будут предметом нашего дальнейшего обсуждения. Но почему? В самом деле, я уже довольно много сказал в подтверждение того, что анионы — это глубокий и зачаровывающий объект. Все это так, но наш курс посвящен квантовым вычислениям, а отнюдь не экзотическим свойствам необычных фаз, реализующихся в конденсированных материальных системах.

Однако, между этими темами имеется тесная связь, впервые по достоинству оцененная Алексеем Китаевым в 1997 году: анионы обеспечивают необычные, захватывающие и, возможно, многообещающие способы реализации отказоустойчивых вычислений.

Так что, похоже, мы должны этим заинтересоваться. Как-никак, я уже прочитал 12 лекций по теории коррекции квантовых ошибок и отказоустойчивых вычислений. Это замечательная теория; я наслаждался, рассказывая о ней, и надеюсь, что вам также доставило удовольствие знакомство с этой теорией. Но она также и обескураживает. Мы видели, что идеальная квантовая схема может быть надежно смоделирована цепью шумящих вентилей, при условии, что шумят они не *слишком* сильно. Мы также видели, что требуемый для успешного моделирования верхний предел размера и глубины схемы вполне приемлем. Эти наблюдения вселяют в нас уверен-

ность, что работоспособные большие квантовые компьютеры когда-нибудь действительно будут построены. И все же, чтобы отказоустойчивость была действительно надежной, квантовые вентили должны иметь достаточно высокую точность воспроизведения (по меркам современной экспериментальной физики), а достижение этого требует значительных накладных расходов. Даже если в принципе надежные квантовые вычисления на шумящих вентилях возможны, никогда не исчезнет стремление повышать точность воспроизведения наших вычислений, совершенствуя, скорее, аппаратное обеспечение, нежели компенсируя его недостатки хитроумными схемными решениями. Используя анионы, можно достичь отказоустойчивости путем разработки «железа» с внутренней сопротивляемостью декогерентизации и другим ошибкам, заметно снижая стремительный рост размеров и глубины моделей наших схем. В таком случае у нас, очевидно, достаточно причин для изучения анионов. Не говоря даже о том, что это будет просто интересно!

В некоторых кругах этот предмет имеет репутацию (на мой взгляд, не совсем заслуженную) трудного и непостижимого. Я намерен начать с основ и не загромождать обсуждение несущественными для наших главных целей деталями. Таким образом, я надеюсь дать ясное представление без доходящих до абсурда упрощений.

Каковы наши цели? Я *не буду* объяснять, как теория анионов связана с наблюдаемыми явлениями в ДКЭХ-системах. В частности, в большинстве этих приложений возникают *абелевы* анионы. С точки зрения квантовой информации, абелевы анионы подходят для надежного *хранения* квантовой информации (и мы уже встречались с первыми признаками такой связи при изучении торических квантовых кодов). Мы обсудим здесь абелевы анионы, но наш основной интерес будет относиться к *неабелевым* анионам, которые, как мы увидим, могут быть наделены неожиданными вычислительными возможностями.

Как было замечено Китаевым [3], система неабелевых анионов с соответствующими свойствами может эффективно моделировать квантовые схемы; его идея была доработана Огберном и мной [4,5] и обобщена Мошоном [6, 7]. В первоначальной схеме Китаева для моделирования некоторого количества квантовых вентилей были необходимы измерения. Фридман, Ларсен и Ванг [11] заметили, что если использовать подходящие для этого анионы, то все измерения можно отложить вплоть до считывания окончательного результата вычисления. Фридман, Китаев и Ванг [10] также показали, что система анионов может эффективно моделироваться квантовой схемой; таким образом, вычислительные возможности анионного квантового компьютера и модели квантовых схем эквивалентны. Цель этих лекций — объяснить эти важные результаты.

Мы сосредоточимся на применении анионов к квантовым вычислениям, оставляя в стороне не менее важную проблему возможной реализации на практике системы анионов с требуемыми свойствами.[1] Я предлагаю вам подумать над этим самостоятельно!

8.2. Композиты поток–заряд

Тем из нас, у кого абстрактные математические конструкции вызывают отвращение, полезно начать изучение теории анионов с размышлений над конкретной моделью. Итак, начнем с напоминания о более знакомом понятии — *эффекте Ааронова–Бома*. Представим электромагнетизм в двумерном мире, где «трубка потока» является локализованным «точечным» объектом (в трех измерениях вы можете представить себе плоскость, пересекаемую перпендикулярным ей магнитным соленоидом). Поток может быть окружен непроницаемой стенкой, так что находящийся снаружи объект не может попасть в область, где магнитное поле отлично от нуля. Но даже в этом случае оно оказывает измеримое влияние на заряженные частицы, находящиеся за пределами трубки потока. Если частица с электрическим зарядом q адиабатически обходит (против часовой стрелки) вокруг потока Φ, ее волновая функция приобретает *топологическую фазу* $e^{iq\Phi}$ (где мы используем единицы, в которых $\hbar = c = 1$). Слово «топологический» здесь означает, что фаза Ааронова–Бома инвариантна относительно непрерывных деформаций траектории заряженной частицы — существенно только «количество оборотов» заряженной частицы вокруг потока.

Концепция топологической инвариантности естественным образом возникает при изучении отказоустойчивости. Топологическими свойствами являются те, что остаются неизменными при непрерывной деформации системы; а отказоустойчивым квантовым вентилем считается тот, чье действие на защищенную информацию остается неизменным (или почти неизменным), когда реализация вентиля деформируется включением шума. Топологическая инвариантность эффекта Ааронова–Бома является важнейшим свойством, которое мы надеемся использовать при разработке внутренне устойчивых квантовых вентилей.

Обычно эффект Ааронова–Бома рассматривается как явление, возникающее в квантовой электродинамике, в которой фотон является строго без-

[1] Недавно в литературе обсуждались два интересных подхода к реализации неабелевых анионов: (1) использование массивов сверхпроводящих контактов и (2) использование холодных атомов, удерживаемых оптическими решетками. [См., например: Л. Б. Иоффе, М. В. Фейгельман, *Реализация топологически защищенных квантовых битов в решетке джозефсоновских контактов*, Успехи физ. наук, **173**, 784–790 (2003). — *Прим. ред.*]

массовым. Однако важно понимать, что явления Аронова – Бома могут случаться и в массивных теориях. Например, мы можем рассмотреть «сверхпроводящую» систему, состоящую из частиц с зарядом e, так что композитные объекты с зарядом ne формируют конденсат (где n — целое число). В этом сверхпроводнике существует квант потока $\Phi_0 = 2\pi/ne$, минимальный ненулевой поток, при обходе вокруг которого частица конденсата, имеющая заряд (ne), приобретает *тривиальную* фазу Аронова – Бома. Изолированная область, содержащая квант потока, представляет собой окруженный сверхпроводящим конденсатом островок нормального металла, защищенный от расползания, поскольку магнитный поток не может проникнуть в сверхпроводник. Это стабильная частица, называемая *флаксоном*. Когда одна из частиц с зарядом e обходит вокруг флаксона, ее волновая функция приобретает нетривиальную топологическую фазу $e^{ie\Phi_0} = e^{2\pi i/n}$. Но в сверхпроводнике фотон приобретает массу благодаря механизму Хиггса, то есть безмассовых частиц здесь нет. Тот факт, что топологические фазы совместимы с массивными теориями, имеет очень важное значение, поскольку легко возбуждаемые безмассовые частицы являются потенциально богатым источником декогерентизации.

Теперь представим, что в нашем двумерном мире поток и электрический заряд намертво связаны друг с другом (по какой-то причине). Флаксон можно представить как поток Φ, захваченный внутри непроницаемого кругового барьера, а электрический заряд q приклеен к *внешней* стороне барьера. Что представляет собой угловой момент этого композита поток–заряд? Предположим, что мы осторожно поворачиваем этот объект на угол 2π против часовой стрелки, возвращая его к первоначальной ориентации. Совершив это, мы перенесли заряд q вокруг потока Φ, породив топологическую фазу $e^{iq\Phi}$. Этот поворот на 2π представляется в гильбертовом пространстве унитарным преобразованием

$$\mathbf{U}(2\pi) = e^{-i2\pi\mathbf{J}} = e^{iq\Phi}, \tag{8.1}$$

где $\mathbf{J}$ — оператор углового момента. Тогда мы приходим к выводу, что возможными собственными значениями углового момента являются

$$J = m - \frac{q\Phi}{2\pi} \quad (m = \text{целое число}). \tag{8.2}$$

Этот спектр можно характеризовать угловой переменной $\theta \in [0, 2\pi)$, определенной как $\theta = q\Phi \pmod{2\pi}$, и говорить, что собственные значения углового момента сдвинуты относительно целых значений на $-\theta/2\pi$. Будем говорить о фазе $e^{i\theta}$, представляющей поворот против часовой стрелки на угол 2π, как о *топологическом спине* композитного объекта.

Но не будет ли поворот на угол 2π действовать на физическую систему тривиально (как если бы ничего не происходило)? Нет, нам это хорошо известно из опыта обращения со *спинорами* в трехмерном пространстве. Для системы, содержащей F фермионов, мы имеем

$$e^{-2\pi i\mathbf{J}} = (-1)^F; \tag{8.3}$$

если количество фермионов нечетно, то собственные значения $\mathbf{J}$ сдвигаются на $1/2$ относительно целых значений. Этот сдвиг физически допустим, поскольку существует *правило суперотбора* по $(-1)^F$: не существует оператора локальной наблюдаемой, способного изменить значение $(-1)^F$ (не существует физического процесса, который может создать или уничтожить изолированный фермион). Действие оператора $e^{-2\pi i\mathbf{J}}$ на когерентную суперпозицию состояний с различными значениями $(-1)^F$ состоит в следующем:

$$e^{-i2\pi\mathbf{J}}(a|\text{ even } F\rangle + b|\text{ odd } F\rangle) = a|\text{ even } F\rangle - b|\text{ odd } F\rangle. \tag{8.4}$$

Относительный знак в суперпозиции обращается, но это не проявляется в наблюдаемых физических эффектах, поскольку все наблюдаемые блочно диагональны в базисе $(-1)^F$.

Аналогично, в двумерии сдвиг спектра углового момента $e^{-2\pi i\mathbf{J}} = e^{i\theta}$ не влечет за собой физически неприемлемых следствий, если существует правило суперотбора по θ, гарантирующее, что относительная фаза в суперпозиции состояний с различными значениями θ физически ненаблюдаема (не только практически, но и в принципе). Как и в случае фермионов, не существует разрешенного физического процесса, способного создать или разрушить изолированный анион.

В трех измерениях допустимы только значения $\theta = 0,\ \pi$ ввиду (как вам, вероятно, известно) топологического свойства трехмерной группы вращений SO(3): замкнутый путь в SO(3), начинающийся из тождественного преобразования и заканчивающийся поворотом на угол 4π, может быть непрерывным образом стянут в тривиальный путь. Отсюда следует, что поворот на угол 4π действительно представляет собой тождественное преобразование и, следовательно, собственными значениями поворота на угол 2π являются $+1$ и -1. Однако группа двумерных вращений SO(2) таким топологическим свойством не обладает, так что, в принципе, возможно любое значение θ.

Заметим, что угловой момент $\mathbf{J}$ изменяет знак при обращении времени ($\mathbf{T}$), а также при отражении ($\mathbf{P}$). За исключением случая, когда θ равен 0 или π, спектр $\mathbf{J}$ асимметричен относительно нуля, и, следовательно, теория анионов обычно неинвариантна относительно $\mathbf{T}$ или $\mathbf{P}$. В модели

композита заряд–поток природа этого нарушения симметрии не составляет загадки — оно происходит от ненулевого магнитного поля. Но если анионы появляются в системе без внутреннего нарушения **T** и **P**, то эти симметрии должны быть нарушены спонтанно или же спектр частиц должен быть «дуальным», так чтобы для каждого аниона с обменной фазой $e^{i\theta}$ существовала идентичная в других отношениях частица с обменной фазой $e^{-i\theta}$.

8.3. Спин и статистика

Для тождественных частиц в трех измерениях существует хорошо известная связь между спином и статистикой: неразличимые частицы с целым спином являются бозонами, а с полуцелым спином — фермионами. В двумерии спин может быть любым вещественным числом. Как эта новая возможность «дробного спина» проявляется в статистике? Ответ в том, что статистика тоже может быть «дробной»!

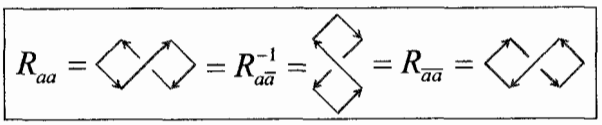
Что происходит, когда мы переставляем два композитных объекта поток–заряд против часовой стрелки? Каждый заряд q адиабатически проходит *половину пути* вокруг потока Ф другого объекта. Тогда можно предположить, что каждый заряд приобретает фазу Аронова–Бома, равную половине фазы, генерируемой при полном обороте заряда вокруг потока. Складывая возникающие от переноса обоих зарядов фазы, мы обнаружим, что перестановка двух композитов поток–заряд изменяет их волновую функцию на фазу

$$\exp\left[i\left(\frac{1}{2}q\Phi+\frac{1}{2}q\Phi\right)\right]=e^{iq\Phi}=e^{i\theta}=e^{-2\pi i\mathrm{J}}. \tag{8.5}$$

Фаза, генерируемая перестановкой двух объектов, совпадает с фазой, генерируемой поворотом одного из них на угол 2π. Таким образом, связь между спином и статистикой сохраняется в виде, который является естественным обобщением связи, применяемой к бозонам и фермионам.

В модели композита поток–заряд природа этой связи довольно прозрачна, но фактически она справедлива в значительно более общем случае. Почему? При чтении учебников по релятивистской квантовой теории поля может легко возникнуть впечатление, что связь между спином и статистикой основывается на лоренцевской инвариантности и имеет отношение к свойствам комплексной группы Лоренца. На самом деле это впечатление обманчиво. Все, что действительно важно для связи между спином и статистикой, — это *существование античастиц*. Специальная относительность — несущественный ингредиент.

Рассмотрим анион, характеризуемый фазой θ, и предположим, что эта частица имеет соответствующую античастицу. Это означает, что частица и ее античастица, комбинируясь, образуют объект с тривиальными квантовыми числами (в частности, с нулевым угловым моментом) и, следовательно, существуют физические процессы, в которых могут рождаться и уничтожаться пары частица–античастица. Проведем в пространстве-времени мировую линию, представляющую процесс, в котором рождаются две пары частица–античастица, одна пара слева, другая — справа (см. рис. ниже).[1] Частица из пары справа переставляется *против часовой стрелки* с частицей из пары слева, после чего обе пары аннигилируют. (Мировая линия имеет направление; если она направлена вперед во времени, то она представляет частицу, а если назад — античастицу.) Поворачивая нашу диаграмму на 90°, мы получаем изображение процесса, в котором рождается отдельная пара частица–античастица, затем частица и античастица переставляются *по часовой стрелке*, после чего пара аннигилирует. Повернем ее на 90° еще раз, тогда мы имеем процесс, в котором рождаются две пары, после чего, прежде чем аннигилировать, *античастица* из правой пары обменивается против часовой стрелки с античастицей из левой пары.



Какой вывод следует из этих манипуляций? Обозначим через R_{ab} унитарный оператор, представляющий перестановку против часовой стрелки двух частиц типа a и b (так что обратный оператор R^{-1}_{ab} представляет перестановку по часовой стрелке), и обозначим через $\bar{a}$ античастицу частицы a. Мы нашли, что

$$R_{aa} = R^{-1}_{a\bar{a}} = R_{\bar{a}\bar{a}}. \tag{8.6}$$

Если a — анион с обменной фазой $e^{i\theta}$, то его античастица $\bar{a}$ имеет *ту же* обменную фазу. Более того, если a и $\bar{a}$ обмениваются против часовой стрелки, то приобретаемая фаза равна $e^{-i\theta}$.

Эти выводы не удивительны, если их интерпретировать на основе модели аниона как композита поток–заряд. Античастица объекта с потоком Ф и зарядом q имеет поток $-\Phi$ и заряд $-q$. Следовательно, когда мы переставляем две античастицы, знаки минус взаимно уничтожаются и результат

[1] На всех трех диаграммах ось времени направлена снизу вверх, то есть изображаемые ими процессы начинаются в крайних нижних точках и заканчиваются в верхних. — *Прим. ред.*

будет тем же, как если бы переставлялись частицы. Но если мы переставляем частицу с античастицей, то относительный знак заряда и потока ведет к появлению обменной фазы $e^{-iq\Phi} = e^{-i\theta}$.

Но в чем состоит связь между этими наблюдениями и соотношением между спином и статистикой? Продолжая созерцать эту пространственно-временную диаграмму, обсудим вытекающие из нее выводы относительно *ориентации* частиц. Чтобы следить за ориентацией, удобно рассматривать мировую линию частицы не как нить, а как *ленту* в пространстве-времени. Я утверждаю, что наш процесс может быть непрерывным образом деформирован к процессу, в котором рождается пара частица–античастица, затем частица поворачивается против часовой стрелки на угол 2π, после чего пара аннигилирует. Удобный способ проверить это утверждение — снять свой ремень (или позаимствовать его у друга). Пряжка на одном конце задает ориентацию; направьте ваш большой палец навстречу пряжке и, прежде чем вновь застегнуть ремень, разверните ее на угол 2π, следуя правилу правой руки. Теперь вы должны быть в состоянии проверить, что можно ориентировать ремень так, чтобы сопоставить ему, а значит и процессу, в котором частица поворачивается на угол 2π, простанственно-временную диаграмму любого из описанных выше процессов перестановок.

Таким образом, в топологическом смысле поворот частицы на угол 2π против часовой стрелки фактически эквивалентен перестановке частиц против часовой стрелки (или перестановке частицы с античастицей по часовой стрелке), что предоставляет удовлетворительное объяснение общей связи между спином и статистикой.[1] Я еще раз подчеркиваю, что в этом рассуждении привлекаются процессы, в которых рождаются и уничтожаются пары частица–античастица, и, следовательно, существование античастиц является важнейшим предварительным условием наличия связи между спином и статистикой.

8.4. Объединение анионов

Мы знаем, что образованный из двух фермионов композитный объект является бозоном. Что произойдет, если построить композитный объект, комбинируя два аниона? Предположим, что a является анионом с обмен-

[1]На самом деле это обсуждение чересчур упрощено. Хотя оно адекватно для абелевых анионов, мы увидим, что для неабелевых анионов его следует усовершенствовать, поскольку в неабелевом случае R_{ab} имеет более одного собственного значения. Аналогично, обсуждение «комбинированных анионов» в следующем разделе также будет необходимо доработать, поскольку в неабелевом случае при соединении двух анионов можно получить более одного типа композитных анионов.

ной фазой $e^{i\theta}$ и что мы образуем «молекулу» из n таких анионов. Какая фаза накапливается при перестановке против часовой стрелки двух таких молекул?

В модели композита поток–заряд ответ очевиден. Каждый из n зарядов одной молекулы, проходя половину пути вокруг каждого из n потоков другой молекулы, приобретает фазу $e^{i\theta/2}$. Тогда всего порождается $2n^2$ фазовых множителей $e^{i\theta/2}$, давая в результате полную фазу

$$e^{i\theta_n} = e^{in^2\theta}. \tag{8.7}$$

Иначе говоря, фаза $e^{i\theta}$ появляется n^2 раз, поскольку на самом деле n анионов одной молекулы переставляются с n анионами другой молекулы. Если бы мы расщепили фермион (скажем) на две идентичные составляющие части, то, вопреки нашим наивным ожиданиям, эти составляющие имели бы обменные фазы $(e^{i\pi})^{1/4} = e^{i\pi/4}$, а не $\sqrt{-1} = i$.

Такое поведение совместимо со связью между спином и статистикой: угловой момент J n-анионной молекулы удовлетворяет условию

$$e^{-2\pi i J_n} = e^{-2\pi i n^2 J} = e^{in^2\theta}. \tag{8.8}$$

Рассмотрим, например, молекулу из двух анионов и представим, что она поворачивается против часовой стрелки на угол 2π. При этом не только каждый анион молекулы поворачивается на угол 2π; кроме этого, один из анионов обходит вокруг другого. Один оборот эквивалентен двум последовательным обменам, так что генерируемая при этом фаза равна $e^{i2\theta}$. Полным результатом двух поворотов и одного оборота является фаза

$$\exp[i(\theta + \theta + 2\theta)] = e^{i4\theta}. \tag{8.9}$$

Почему угловые моменты анионов комбинируются неаддитивно, можно понять и другим способом, заметив, что полный угловой момент молекулы состоит из двух частей — спинового углового момента S каждого из двух анионов (который *является* аддитивным) и *орбитального* углового момента L анионной пары. Поскольку перенос против часовой стрелки одного аниона вокруг другого порождает нетривиальную фазу $e^{i2\theta}$, зависимость двуханионной волновой функции ψ от относительного азимутального угла φ неоднозначна; вместо этого имеет место соотношение

$$\psi(\varphi + 2\pi) = e^{-i2\theta}\psi(\varphi). \tag{8.10}$$

Это означает, что спектр орбитального углового момента $\mathbf{L}$ сдвинут относительно целых значений:

$$e^{-i2\pi\mathbf{L}} = e^{2i\theta}, \tag{8.11}$$

и этот орбитальный угловой момент аддитивно комбинируется со спином S, в результате чего полный угловой момент становится равным

$$-2\pi J = -2\pi L - 2\pi S = 2\theta + 2\theta + 2\pi \cdot (\text{целое число}) = \\ = 4\theta + 2\pi \cdot (\text{целое число}). \quad (8.12)$$

Что если, с другой стороны, мы построим молекулу $\bar{a}a$ из аниона a и его античастицы $\bar{a}$? Тогда, как мы видели, спин S имеет то же самое значение, как и для молекулы aa. Но обменная фаза имеет противоположное значение, так что нецелая часть орбитального углового момента равна $-2\pi L = -2\theta$ вместо $-2\pi L = 2\theta$, а полный угловой момент $J = L + S$ является целым. Конечно, это свойство необходимо, для того чтобы пара $\bar{a}a$ могла аннигилировать, не оставляя после себя объекта, несущего нетривиальный угловой момент.

8.5. Унитарные представления группы «кос»

Мы уже отмечали, что спектр углового момента в двумерном пространстве обладает иными, нежели в трех измерениях, свойствами, поскольку SO(2) имеет другие по сравнению с SO(3) топологические свойства [(SO(3) имеет компактную односвязную накрывающую группу SU(2), тогда как SO(2) — нет]. Это наблюдение дает нам еще одну возможность понять, почему анионы возможны в двух измерениях и невозможны в трех. Также поучительно заметить, что *перестановка* частиц в двух и трех пространственных измерениях имеет разные топологические свойства.

Как мы обнаружили в нашем обсуждении связи между статистикой частиц и античастиц, перестановку частиц полезно рассматривать как процесс, протекающий в пространстве-времени. В частности, полезно представлять, что мы находим амплитуду квантового перехода для зависящего от времени n-частичного процесса, вычисляя сумму по историям частиц (хотя для наших целей вряд ли потребуется вычислять какие-либо интегралы по траекториям).

Рассмотрим систему n неразличимых точечных частиц, удерживаемых на двумерной поверхности (которую пока можно считать плоской), и предположим, что никакие две частицы не могут занимать одно и то же положение. Мы можем рассматривать конфигурацию частиц в фиксированный момент времени как плоскость с n «проколами» в указанных положениях, то есть с каждой частицей на поверхности связывается дырка с бесконечно малым радиусом. Условие, что частицам запрещено находиться в совпадающих позициях, обеспечивается тем, что в каждый момент времени

в плоскости существует точно n проколов. Более того, коль скоро частицы неразличимы, каждый прокол в точности такой же, как и любой другой. Таким образом, если мы произвели перестановку n проколов, это не повлечет за собой никаких физических эффектов; все проколы одинаковы, так что «кто есть кто» здесь совершенно не важно. Все, что действительно имеет значение, это n различных положений частиц на плоскости.

Чтобы вычислить квантовую амплитуду эволюции n частиц из конфигурации, задаваемой начальными положениями в момент времени $t = 0$, в конфигурацию, задаваемую их конечными положениями в момент времени $t = T$, необходимо просуммировать по всем классическим историям этих n частиц между начальной и конечной конфигурациями, взвешенным фазами e^{iS}, где S — классическое действие истории. Если мы рассматриваем мировую линию каждой частицы как нить, то каждую историю n частиц можно представить в виде «косы», в которой каждая частица начального ($t = 0$) временного среза может быть связана нитью с любой из частиц конечного ($t = T$) временного среза. Более того, поскольку пересечение мировых линий частиц запрещено, множество кос распадается на различные топологические классы, которые невозможно непрерывным образом деформировать друг в друга, а интеграл по траекториям разлагается на сумму вкладов, вносимых историями различных топологических классов.

Нетривиальные операции перестановок, действующие на частицы конечного временного среза, изменяют топологический класс косы. Таким образом, мы видим, что элементы группы симметрии, генерируемой перестановками, находятся во взаимно однозначном соответствии с топологическими классами. Эта (бесконечная) группа B_n называется группой кос из n нитей; групповой композиционный закон соответствует сочленению кос (то есть соединению следующих друг за другом кос). В квантовой теории состояние n неразличимых частиц принадлежит гильбертовому пространству, которое преобразуется по унитарному представлению группы кос B_n.

Группа может быть представлена набором генераторов, подчиняющихся особым определяющим соотношениям. Чтобы понять определяющие соотношения, представим, что n частиц занимают n распределенных на линии упорядоченных позиций (помеченных как $1, 2, 3, \ldots, n$). Пусть σ_1 обозначает перестановку против часовой стрелки частиц, первоначально занимавших позиции 1 и 2, σ_2 обозначает перестановку против часовой стрелки частиц, первоначально занимавших позиции 2 и 3, и так далее. Любая коса может быть построена как последовательность перестановок соседних частиц; следовательно, $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ являются генераторами группы.

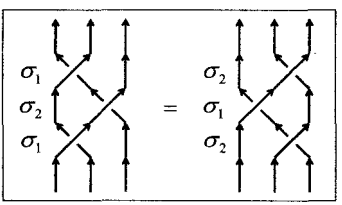
Имеется два типа определяющих соотношений, которым удовлетворяют эти генераторы. Соотношение первого типа

$$\sigma_j\sigma_k = \sigma_k\sigma_j, \quad |j-k| \geqslant 2, \tag{8.13}$$

утверждает лишь то, что перестановки несоседних пар частиц коммутируют. Соотношением второго, несколько более тонкого, типа является

$$\sigma_j\sigma_{j+1}\sigma_j = \sigma_{j+1}\sigma_j\sigma_{j+1}, \quad j = 1, 2, \ldots, n-2, \tag{8.14}$$

которое иногда называют *соотношением Янга–Бэкстера*. Вы можете проверить его, изобразив на листе бумаги две косы $\sigma_1\sigma_2\sigma_1$ и $\sigma_2\sigma_1\sigma_2$ и заметив, что обе они описывают процесс, в котором частицы, изначально находившиеся в позициях 1 и 3, переставляются против часовой стрелки вокруг частицы 2, которая остается неподвижной, то есть это топологически эквивалентные косы.



Поскольку группа кос бесконечна, она имеет бесконечное количество неприводимых унитарных представлений, а фактически существует бесконечное число *одномерных* представлений. Неразличимые частицы, преобразующиеся по одномерному представлению группы кос, называются *абелевыми анионами*. В одномерных представлениях каждый генератор σ_j группы B_n представлен фазой $\sigma_j = e^{i\theta_j}$. Более того, соотношение Янга–Бэкстера принимает вид $e^{i\theta_j}e^{i\theta_{j+1}}e^{i\theta_j} = e^{i\theta_{j+1}}e^{i\theta_j}e^{i\theta_{j+1}}$, откуда следует, что $e^{i\theta_j} = e^{i\theta_{j+1}} \equiv e^{i\theta}$ — все перестановки представляются *одной и той же* фазой. Конечно, смысл этого равенства в том, что обменные фазы не должны зависеть от того, какие пары переставляются, если частицы действительно неразличимы. При $\theta = 0$ мы получаем бозоны, а при $\theta = \pi$ — фермионы.

Группа кос также имеет множество неабелевых представлений, размерность которых больше единицы; неразличимые частицы, преобразующиеся по этим представлениям, называются *неабелевыми анионами* (или иногда *неабелионами*). Чтобы понять физические свойства неабелевых анионов, нам понадобится разобраться в математической структуре некоторых

из этих представлений. В ходе лекций я надеюсь передать некоторое интуитивное понимание неабелевых анионов, детально обсудив некоторые примеры.

А пока можно уже предугадать главную задачу, которую мы надеемся решить. Для неабелевых анионов реализованное n анионами неприводимое представление группы B_n действует на «топологическом векторном пространстве» V_n, размерность которого D_n экспоненциально растет вместе с n. Для анионов с подходящими свойствами образ представления может быть *плотным* в $\mathrm{SU}(D_n)$. Тогда сплетение анионов может моделировать квантовое вычисление, то есть подходящим выбором косы можно со сколь угодно высокой точностью воспроизведения реализовать любое (частное) унитарное преобразование, действующее в экспоненциально большом векторном пространстве V_n.

Таким образом, нас очень интересуют неабелевы представления группы кос. Но следует также подчеркнуть (а ниже мы обсудим это подробнее), что это больше относится к модели анионов, нежели просто к представлению группы кос. В модели трубки потока для абелевых анионов мы были в состоянии описывать не только результаты перестановок анионов, но также и типы частиц, которые можно получить, комбинируя два или больше анионов. Аналогично, общая анионная модель, описывающая различные типы анионов, включает в себя «правила композиции», которые определяют, какие типы анионов могут быть получены путем комбинации двух конкретных типов анионов. Нетривиальные условия согласования возникают вследствие ассоциативности соединения (слияние a с b с последующим соединением результата с c эквивалентно слиянию b с c с последующим соединением результата с a), а также вследствие того, что правила композиции (или слияния) должны согласовываться с правилами сплетения. Несмотря на то, что эти условия согласования накладывают жесткие ограничения, существует множество решений и, следовательно, множество в принципе реализуемых моделей неабелевых анионов.

8.6. Топологическое вырождение

Прежде чем перейти к неабелевым анионам, нам следует обсудить еще одно важное понятие, касающееся абелевых анионов. В любой модели анионов (в сущности, в любой локальной квантовой системе с массовой щелью) существует *основное*, или *вакуумное, состояние* — состояние, в котором отсутствуют частицы. Основное состояние на плоскости единственно, но в случае двумерной поверхности с нетривиальной топологией оно вырожденно, причем степень вырождения зависит от топологии. Мы уже

встречались с явлением «топологического вырождения» в модели абелевых анионов при изучении особого квантового кода коррекции ошибок, торического кода Китаева. Теперь мы увидим, что топологическое вырождение является общим свойством любой модели (абелевых) анионов.

К понятию топологического вырождения можно подойти, исследуя представления простой операторной алгебры. Рассмотрим случай тора, представляемого как квадрат с тождественными противоположными сторонами, и рассмотрим два фундаментальных 1-цикла на торе: C_1, который наматывается на квадрат в направлении x_1, и C_2, который наматывается в направлении x_2. Можно построить унитарный оператор $\mathbf{T}_1$, описывающий процесс, в котором рождается пара анион–антианион, анион распространяется вдоль C_1, после чего пара аннигилирует. Аналогично можно построить унитарный оператор $\mathbf{T}_2$, описывающий процесс, в котором рождается пара, и прежде чем она аннигилирует, анион распространяется вдоль цикла C_2. Каждый из операторов $\mathbf{T}_1$ и $\mathbf{T}_2$ сохраняет основное состояние системы (состояние в отсутствие частиц); действительно, каждый из них коммутирует с гамильтонианом $\mathbf{H}$ системы и, следовательно, может быть диагонализован одновременно с $\mathbf{H}$ ($\mathbf{T}_1$ и $\mathbf{T}_2$ являются симметриями).

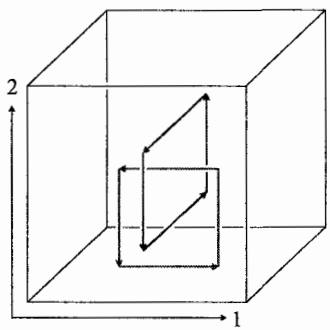
Однако $\mathbf{T}_1$ и $\mathbf{T}_2$ не коммутируют друг с другом. Если наш тор имеет бесконечный пространственный объем и существует массовая щель (так что взаимодействие между пространственно разделенными анионами возникает только благодаря эффекту Аронова–Бома), то коммутатор $\mathbf{T}_1$ и $\mathbf{T}_2$ равен[1]

$$\mathbf{T}_2^{-1}\mathbf{T}_1^{-1}\mathbf{T}_2\mathbf{T}_1 = e^{-i2\theta}\mathbf{1}, \tag{8.15}$$

где $e^{i\theta}$ — обменная фаза аниона. Нетривиальный коммутатор появляется, поскольку процесс (в котором (1) анион обходит вокруг C_1, (2) анион обходит вокруг C_2, (3) анион обходит вокруг C_1 в обратном направлении и (4) анион обходит вокруг C_2 в обратном направлении), топологически эквивалентен обходу по часовой стрелке одного аниона вокруг другого. Чтобы проверить это утверждение, рассмотрим действие $\mathbf{T}_2^{-1}\mathbf{T}_1^{-1}\mathbf{T}_2\mathbf{T}_1$ как процесс в пространстве-времени. Заметим сначала, что процесс, описываемый оператором $\mathbf{T}_1^{-1}\mathbf{T}_1$, в котором мировая линия аниона проходит сначала через C_1 и непосредственно после этого проходит C_1 в обратном направлении, может быть деформирован в процесс, в котором мировая линия аниона обходит топологически тривиальную петлю, которую можно непрерывным образом стянуть в точку (в соответствии с тем, что $\mathbf{T}_1^{-1}\mathbf{T}_1$ в действительности является единичным оператором). Подобным образом процесс, описы-

[1]Здесь используется алгебраическое определение коммутатора $[\mathbf{A}, \mathbf{B}] = \mathbf{A}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{-1}$. — *Прим. ред.*

ваемый оператором $\mathbf{T}_2^{-1}\mathbf{T}_1^{-1}\mathbf{T}_2\mathbf{T}_1$, может быть деформирован к процессу, в котором мировые линии анионов обходят две замкнутые петли, но таким образом, что они один раз *сцепляются* друг с другом (см. рис. ниже); более того, одна петля протыкает поверхность, ограниченную другой петлей, в направлении, противоположном ее ориентации, определяемой правилом правой руки. Этот процесс можно непрерывным образом деформировать к процессу, в котором рождаются две пары, один анион обходит другой по часовой стрелке, а затем обе пары аннигилируют. Обход по часовой стрелке эквивалентен двум последовательным перестановкам по часовой стрелке, что в нашем одномерном представлении группы кос представляется фазой $e^{-i2\theta}$. Таким образом, $\mathbf{T}_1$ и $\mathbf{T}_2$ не коммутируют за исключением случаев $\theta = 0$ (бозоны) и $\theta = \pi$ (фермионы).



Поскольку $\mathbf{T}_1$ и $\mathbf{T}_2$ коммутируют с гамильтонианом $\mathbf{H}$, они оба сохраняют собственные подпространства $\mathbf{H}$, но поскольку $\mathbf{T}_1$ и $\mathbf{T}_2$ не коммутируют между собой, они не могут быть одновременно диагонализованы. Так как $\mathbf{T}_1$ – унитарный оператор, его собственные значения представляют собой фазы; используем угловую переменную $\alpha \in [0, 2\pi)$ для обозначения собственного состояния оператора $\mathbf{T}_1$ с собственным значением $e^{i\alpha}$:

$$\mathbf{T}_1|\alpha\rangle = e^{i\alpha}|\alpha\rangle. \tag{8.16}$$

Тогда действие оператора $\mathbf{T}_2$ на собственное состояние оператора $\mathbf{T}_1$ повышает значение α на 2θ:

$$\mathbf{T}_1(\mathbf{T}_2|\alpha\rangle) = e^{i2\theta}\mathbf{T}_2\mathbf{T}_1|\alpha\rangle = e^{i2\theta}e^{i\alpha}(\mathbf{T}_2|\alpha\rangle). \tag{8.17}$$

Предположим, что θ — *рациональное* кратное 2π, которое можно представить в виде

$$\theta = \pi p/q, \tag{8.18}$$

где q и p ($p < 2q$) — взаимно простые положительные целые числа. Тогда мы приходим к выводу, что $\mathbf{T}_1$ должен иметь по крайней мере q различных собственных значений; $\mathbf{T}_1$, действуя на $|\alpha\rangle$, генерирует орбиту с q различными значениями

$$\alpha + \left(\frac{2\pi p}{q}\right) k \pmod{2\pi}, \quad k = 0, 1, 2, \ldots, q-1. \tag{8.19}$$

Так как $\mathbf{T}_1$ коммутирует с $\mathbf{H}$, основное состояние нашей анионной системы на торе (по сути, собственное состояние с любой энергией) должно иметь вырождение, являющееся целым кратным числа q. Действительно, в общем случае (без учета дополнительных симметрий или случайных вырождений) кратность вырождения ожидается в точности равной q.

Для двумерной поверхности рода g (сферы с g «ручками») кратность топологического вырождения равна q^g, поскольку существуют аналогичные $\mathbf{T}_1$ и $\mathbf{T}_2$ операторы, связанные каждой из g ручек, и все операторы типа $\mathbf{T}_1$ могут быть одновременно диагонализованы. Более того, аналогичное рассуждение можно применить и к конечной плоской системе, если на ее *границах* могут рождаться и уничтожаться *отдельные* анионы. Рассмотрим, например, кольцо, на внутренней и внешней границах которого могут появляться и исчезать анионы. Тогда мы можем характеризовать унитарный оператор $\mathbf{T}_1$ как описывающий процесс, в котором анион обходит кольцо против часовой стрелки, а унитарный оператор $\mathbf{T}_2$ — как описывающий процесс, в котором анион появляется на внешней границе, распространяется к внутренней границе и там исчезает. Эти операторы $\mathbf{T}_1$ и $\mathbf{T}_2$ имеют такой же коммутатор, что и соответствующие операторы, определенные на торе. Таким образом, мы снова приходим к выводу, что при $\theta = \pi p/q$ основное состояние на кольце q-кратно вырожденно. Для диска с h дырками существует аналогичный $\mathbf{T}_1$ оператор, который оборачивает анион против часовой стрелки вокруг каждой дырки, и аналогичный $\mathbf{T}_2$ оператор, который переносит анион от внешней границы диска к границе дырки; таким образом, кратность вырождения равна q^h.

То, что мы здесь описали, представляет собой жесткую *топологическую квантовую память*. Фаза $e^{i2\theta} = e^{i2\pi p/q} \equiv \omega$, приобретаемая при обращении против часовой стрелки одного аниона вокруг другого, является первообразным q-м корнем из единицы, и в случае плоской системы с дырками оператор $\mathbf{T}_1$ может рассматриваться как закодированный оператор Паули $\bar{\mathbf{Z}}$, действующий на ассоциированную с данной дыркой q-мерную систему. С физической точки зрения, собственное значение ω^s оператора $\bar{\mathbf{Z}}$ лишь подсчитывает количество s анионов, «упакованных» внутри дырки.

Оператор $\mathbf{T}_2$ может рассматриваться как дополнительный оператор Паули $\bar{\mathbf{X}}$, увеличивающий значение s, переводя один анион от границы системы и помещая его в дырке. Поскольку квантовая информация закодирована в нелокальном свойстве системы, она хорошо защищена от декогерентизирующего действия окружения. Подобное помещение квантового состояния в память и его считывание может быть многообещающим для такой системы, так как, по крайней мере в принципе, $\bar{\mathbf{Z}}$ можно было бы измерить, скажем, выполняя интерференционный эксперимент, в котором налетающий анион рассеивается на дырке. Позднее мы увидим, что, используя неабелевы анионы, можно упростить считывание; кроме того, с неабелевыми анионами топологическими свойствами можно пользоваться как для *обработки* квантовой информации, так и для ее хранения.

Насколько устойчивой является эта квантовая память? Нас должны беспокоить ошибки, возникающие вследствие тепловых и квантовых флуктуаций. Тепловые флуктуации способны вызывать рождение анионов, которые могут диффундировать в окрестности одной из дырок в образце или от одной границы до другой, являясь причиной ошибок кодирования. Тепловые ошибки сильно подавляются больцмановским множителем $e^{-\Delta/T}$, если температура T достаточно мала по сравнению с энергетической щелью Δ (минимальной энергией, необходимой для рождения одного аниона на границе образца или анион-антианионной пары в объеме). Вредными квантовыми флуктуациями являются процессы туннелирования, в которых возникает виртуальная анион-антианионная пара и, прежде чем аннигилировать, анион распространяется вокруг дырки или виртуальный анион появляется на границе дырки и, прежде чем исчезнуть, распространяется до другой границы. Эти обусловленные квантовым туннелированием ошибки сильно подавляются, если дырки достаточно велики и достаточно далеко отделены друг от друга и от внешних границ.[1]

Заметим, что наш вывод о конечности топологического вырождения зависит от предположения о том, что угол θ является рациональным кратным π. Можно сказать, что теория анионов *рациональна*, если топологическое вырождение конечно для любой поверхности конечного рода (а для неабелевых анионов — если топологическое векторное пространство V_n конечномерно для любого конечного числа анионов n). Можно ожидать, что анионы, возникающие в любой физически разумной системе, будут в этом

[1]Если вы знакомы с методами евклидового интегрирования по траекториям, вы сможете просто проверить, что в главном полуклассическом приближении амплитуда A такого процесса туннелирования, в котором анион распространяется на расстояние L, имеет вид $A = Ce^{-L/L_0}$, где C — константа, а $L_0 = \hbar(2m^*\Delta)^{-1/2}$; здесь $\hbar$ — постоянная Планка, а m^* — эффективная масса аниона, определяемая таким образом, что кинетическая энергия аниона, движущегося со скоростью v, равна $m^*v^2/2$.

смысле рациональными, и, таким образом, следует ожидать, что они будут иметь обменные фазы, являющиеся корнями из единицы.

8.7. Еще раз о торических кодах

Если эти замечания о топологическом вырождении кажутся до боли знакомыми, то, возможно, потому, что аналогичные аргументы мы использовали при обсуждении торических кодов.

Торический код можно рассматривать как (вырожденное) основное состояние системы кубитов, заполняющих ребра квадратной решетки на торе, с гамильтонианом

$$\mathbf{H} = -\frac{1}{4}\Delta\left(\sum_P \mathbf{Z}_P + \sum_S \mathbf{X}_S\right), \tag{8.20}$$

здесь плакетный[1] оператор $\mathbf{Z}_P = \otimes_{\ell\in P}\mathbf{Z}_\ell$ представляет собой тензорное произведение операторов $\mathbf{Z}$, действующих на четыре кубита, расположенных на ребрах плакета P, а узельный оператор $\mathbf{X}_S = \otimes_{\ell\ni S}\mathbf{X}_\ell$ — тензорное произведение операторов $\mathbf{X}$, действующих на четыре кубита, расположенных на ребрах, сходящихся в узле S. Плакетные и узельные операторы являются в точности (коммутирующими) генераторами стабилизатора торического кода. Основное состояние одновременно является собственным состоянием всех генераторов стабилизатора с собственным значением $+1$.

В этой модели существует два типа возбуждений локализованных квазичастиц: плакетные возбуждения с $\mathbf{Z}_P = -1$,[2] которые можно представлять как магнитные флаксоны (кванты магнитного потока), и узельные возбуждения с $\mathbf{X}_S = -1$, которые можно толковать как электрические заряды. Действующая на ребре Z-ошибка порождает пару зарядов на двух связанных этим ребром узлах, тогда как действующая на ребре X-ошибка порождает пару флаксонов на двух разделенных этим ребром плакетах. Энергетическая щель Δ представляет собой плату за рождение пар любого из двух типов.

Заряды являются бозонами по отношению друг к другу (они имеют тривиальную обменную фазу $e^{i\theta} = 1$), флаксоны также являются бозонами относительно друг друга. Поскольку флаксоны отличимы от зарядов,

[1] Plaquette (франц.) — «плакет»; буквально: небольшая металлическая пластинка. В данном случае — элементарный квадрат решетки. Термин позаимствован из квантовой теории калибровочных полей на решетках. — *Прим. ред.*

[2] Это и следующее за ним аналогичное выражение представляют собой символическую запись того, что данное элементарное возбуждение является собственным состоянием оператора $\mathbf{Z}_P$ или $\mathbf{X}_S$ с собственным значением -1. — *Прим. ред.*

не имеет смысла рассматривать перестановки между ними. Но фаза (-1), приобретаемая при обходе заряда вокруг потока, делает эту модель анионной. Кратность вырождения основного состояния (размерность кодового пространства) можно интерпретировать как следствие этого свойства частиц.

Поскольку в этой модели на торе существует два типа частиц, существует и два типа операторов $\mathbf{T}_1$: оператор $\mathbf{T}_{1,S}$, который переносит заряд (узельный дефект) по 1-циклу C_1, и оператор $\mathbf{T}_{1,P}$, который переносит флаксон (плакетный дефект) по C_1. Аналогично, существует два типа операторов $\mathbf{T}_2$: операторы $\mathbf{T}_{2,S}$ и $\mathbf{T}_{2,P}$. Оба нетривиальных коммутатора

$$\mathbf{T}_{2,P}^{-1}\mathbf{T}_{1,S}^{-1}\mathbf{T}_{2,P}\mathbf{T}_{1,S} = -1 = \mathbf{T}_{2,S}^{-1}\mathbf{T}_{1,P}^{-1}\mathbf{T}_{2,S}\mathbf{T}_{1,P} \tag{8.21}$$

возникают в процессах, в которых мировые линии зарядов и флаксонов зацепляются только один раз. Таким образом, $\mathbf{T}_{1,S}$ и $\mathbf{T}_{2,S}$ могут быть одновременно диагонализованы и могут рассматриваться как закодированные операторы Паули $\bar{\mathbf{Z}}_1$ и $\bar{\mathbf{Z}}_2$, действующие на два защищаемых кубита. Оператор $\mathbf{T}_{2,P}$, коммутирующий с $\bar{\mathbf{Z}}_1$ и антикоммутирующий с $\bar{\mathbf{Z}}_2$, может рассматриваться как закодированный оператор $\bar{\mathbf{X}}_1$, и аналогично $\mathbf{T}_{1,P}$ представляет собой закодированный оператор $\bar{\mathbf{X}}_2$.

Для сконструированного нами на торе идеального гамильтониана вырождение четырех основных состояний является точным (частицы имеют бесконечные эффективные массы). Слабые локальные возмущения будут снимать вырождение, но только на величину, экспоненциально убывающую с ростом линейного размера тора L. Чтобы быть конкретнее, предположим, что возмущение представляет собой направленное вдоль оси $\hat{z}$ однородное «магнитное поле», взаимодействующее с магнитными моментами кубитов,

$$\mathbf{H}' = -h\sum_{\ell}\mathbf{Z}_{\ell}. \tag{8.22}$$

Поскольку энергетическая щель отлична от нуля, для вычисления главного вклада в расщепление вырождения по теории возмущений достаточно рассматривать действие возмущения в четырехмерном подпространстве, натянутом на основные состояния невозмущенной системы. В торическом коде операторами с нетривиальными матричными элементами в этом подпространстве являются те, чьи сомножители $\mathbf{Z}_{\ell}$ действуют на ребрах, образующих замкнутые петли, которые обходят вокруг тора (или чьи сомножители $\mathbf{X}_{\ell}$ действуют на ребрах, дуальных ребрам, которые образуют замкнутые петли, обходящие вокруг тора). Для решетки размера $L \times L$ на торе минимальная длина такой замкнутой петли равна L; следовательно, отличные от

нуля матричные элементы возникают, начиная лишь с L-го порядка теории возмущений, и подавляются множителем h^L. Таким образом, при малых h и больших L ошибки памяти, вызванные квантовыми флуктуациями, возникают только с экспоненциально малой амплитудой вероятности.

8.8. Неабелев эффект Аронова – Бома

Существует красивая абстрактная теория неабелевых анионов, и в свое время мы немного в ней покопаемся. Но я предпочел бы приступить к изучению этого предмета с описания более конкретной модели.

Имея в виду эту цель, вспомним некоторые особенности *хромодинамики*, теории кварков и глюонов, содержащихся внутри атомных ядер и других сильно взаимодействующих частиц. В реальном мире кварки постоянно связаны друг с другом и никогда не встречаются в свободном виде. Однако для нашего обсуждения представим фантастический мир, в котором силы между кварками настолько слабы, что характерный масштаб расстояний конфайнмента кварков очень велик.

Кварки имеют степень свободы, которую на образном языке называют *цветом*. Существует три типа кварков, которые, пользуясь этой метафорой, мы называем красным («red», R), желтым («yellow», Y) и синим («blue», B). Кварки всех трех цветов физически идентичны, и лишь когда мы сводим их вместе, можно говорить о том, являются ли их цвета одинаковыми (взаимодействие между одинаковыми цветами имеет характер отталкивания) или различными (разные цвета притягиваются друг к другу). Нет ничего, что помешало бы мне создать в моей лаборатории *кварковое бюро стандартов*, где раскрашенные кварки сортируются по трем корзинам; все кварки, лежащие в одной корзине, имеют одинаковые цвета, а кварки из разных корзин — разные. Мы можем (произвольным образом) снабдить три корзины метками — R, Y, B.

Если, прогуливаясь за пределами лаборатории, я обнаружу ранее не замеченный кварк, то на первых порах я буду неуверен относительно его цвета. Но это можно выяснить. Я захватываю кварк с собой и осторожно, чтобы по дороге не разрушить его цвет (в хромодинамике существует понятие *параллельного переноса* цвета), возвращаюсь в свою лабораторию. Вернувшись в кварковое бюро стандартов, я могу сравнить этот новый кварк с ранее калиброванными кварками в корзинах и определить, как должен быть помечен новый кварк: R, Y или B.

Это звучит просто, но есть одна загвоздка: в хромодинамике параллельный перенос цвета *зависит от пути*, благодаря влияющему на цвет

эффекту Ааронова–Бома. Допустим, что в кварковом бюро стандартов приготовлен кварк, цвет которого описывается квантовым состоянием

$$|\psi_q\rangle = q_R|R\rangle + q_Y|Y\rangle + q_B|B\rangle; \tag{8.23}$$

это когерентная суперпозиция с амплитудами q_R, q_Y и q_B для красного, желтого и синего состояний. Кварк движется вдоль пути, который обходит *трубку цветового магнитного потока* и возвращается в кварковое бюро стандартов, где его цвет может быть снова калиброван. По возвращении его цветовое состояние оказывается повернутым

$$\begin{pmatrix} q'_R \\ q'_Y \\ q'_B \end{pmatrix} = \mathbf{U} \begin{pmatrix} q_R \\ q_Y \\ q_B \end{pmatrix}, \tag{8.24}$$

где $\mathbf{U}$ — (специальная) унитарная 3×3-матрица. Подобно этому, если вновь обнаруженный кварк будет перенесен в бюро стандартов, результат измерения его цвета будет зависеть от того, обошел ли он во время своего вояжа трубку потока слева или справа.

Эта зависимость от пути параллельного переноса цвета очень похожа на зависимость от пути параллельного переноса касательного вектора на искривленном римановом многообразии. В хромодинамике магнитное поле представляет собой *кривизну*, величина которой определяет степень зависимости от пути.

В общем случае SU(3)-матрица $\mathbf{U}$, описывающая результат параллельного переноса цвета вдоль замкнутого пути, зависит от *стартовой точки* x_0, в которой этот путь начинается и заканчивается, а также и от замкнутой петли C, которую он описывает. В тех случаях, когда важно указывать петлю и базисную точку, мы будем использовать обозначение $\mathbf{U}(C, x_0)$. Собственные числа матрицы $\mathbf{U}$ имеют инвариантный «геометрический» смысл, характеризующий параллельный перенос, но сама $\mathbf{U}$ зависит от соглашений, принятых нами в стартовой точке. Возможно, вы предпочтете выбрать отличный от принятого мной ортонормированный базис для пространства цветов в стартовой точке x_0, так что ваши стандартные цвета R, Y и B будут отличаться от моих действием SU(3)-матрицы $\mathbf{V}(x_0)$. Тогда, если я описываю влияние параллельного переноса вокруг петли C матрицей $\mathbf{U}$, вы характеризуете его другой матрицей

$$\mathbf{V}(x_0)\mathbf{U}(C, x_0)\mathbf{V}^{-1}(x_0), \tag{8.25}$$

отличающейся от моей сопряжением матрицей $\mathbf{V}(x_0)$. Физики иногда говорят о свободе переопределения соглашений как о выборе *калибровки* и го-

ворят, что сама $\mathbf{U}$ *зависит от калибровки*, тогда как ее собственные числа — *калибровочно инвариантны*.

Хромодинамика на рассматриваемых здесь масштабах (гораздо меньших характерной длины конфайнмента кварков) представляет собой теорию типа электродинамики с дальнодействующими кулоновскими взаимодействиями между кварками, переносимыми посредством «глюонных» полей. Мы предпочтем рассматривать теорию, которая сохраняет некоторые черты хромодинамики (в частности, зависимость от пути переноса цвета), но без легко возбуждаемых легких глюонов. В случае электродинамики мы исключали легкие фотоны, рассматривая «сверхпроводник», в котором заряженные частицы образуют конденсат, магнитные поля вытеснены, а магнитный поток изолированного объекта квантуется. Обратимся к этой идее и здесь. Рассмотрим *неабелев сверхпроводник* в двух пространственных измерениях. Этот мир содержит частицы, несущие «кванты магнитного потока» (подобные квантам цветового магнитного потока в хромодинамике), и частицы, несущие заряд (подобные цветным кваркам хромодинамики). Поток принимает значения в *конечной неабелевой группе* G, а зарядам соответствуют *унитарные неприводимые представления* группы G. В такой постановке можно сформулировать некоторые интересные модели неабелевых анионов.

Пусть R обозначает конкретное неприводимое представление группы G, размерность которого обозначается как $|R|$. Мы можем открыть «зарядовое бюро стандартов» и определить произвольно выбранный ортонормированный базис в $|R|$-мерном векторном пространстве, в котором действует R:

$$|R,i\rangle, \quad i=1,2,\ldots|R|. \tag{8.26}$$

Если заряд R обходит замкнутый путь, окружающий поток $a \in G$, то возникает нетривиальный эффект Ааронова – Бома — базис для R поворачивается унитарной матрицей $\mathbf{D}^R(a)$, представляющей a:

$$|R,j\rangle \longmapsto \sum_{i=1}^{|R|} |R,i\rangle D^R_{ij}(a). \tag{8.27}$$

В принципе, матричные элементы $D^R_{ij}(a)$ измеримы, например, в интерференционных экспериментах, в которых пучок калиброванных зарядов может пройти по любую сторону от кванта потока. (Фаза комплексного числа $D^R_{ij}(a)$ определяет величину *сдвига* интерференционных полос, а модуль $D^R_{ij}(a)$ — их *контрастность*.) Таким образом, как только для зарядов выбран стандартный базис, их можно использовать для того, чтобы присвоить метки (элементы G) всем квантам потока. Это соответствие однозначно,

коль скоро представление R является точным и исключены любые групповые автоморфизмы (порождающие неоднозначности, которые мы свободны разрешать по своему усмотрению).

Однако групповые элементы, которые мы сопоставляем квантам потока, зависят от наших соглашений. Допустим, что мне предоставили k флаксонов (частиц, несущих кванты потока), и я использую мои стандартные заряды для измерения кванта потока каждой частицы. Я ставлю в соответствие этим k флаксонам элементы группы $a_1, a_2, \ldots, a_k \in G$. Затем прошу вас измерить квант потока, чтобы проверить мое распределение. Но ваши стандартные заряды отличаются от моих, вследствие того, что их незаметно перенесли вокруг другого потока (который я бы обозначил как $g \in G$). Следовательно, этим же k флаксонам вы сопоставите элементы группы $ga_1g^{-1}, ga_2g^{-1}, \ldots, ga_kg^{-1}$; то есть наши сопоставления отличаются сопряжением по элементу g.

Урок, извлекаемый из этого примера, состоит в том, что сопоставление флаксонам групповых элементов по сути неоднозначно и не имеет инвариантного смысла. Но поскольку правильные соответствия между элементами группы и флаксонами отличаются лишь сопряжением по некоторому элементу $g \in G$, действительно инвариантный смысл, с которым будут согласны все наблюдатели, имеют *классы сопряженных элементов*, соответствующих квантам потока в G. Действительно, даже если мы зафиксируем наши соглашения в зарядовом бюро стандартов, сопоставляемый конкретному флаксону групповой элемент может измениться, если этот флаксон примет участие в физическом процессе, в котором он сплетется с другими флаксонами. По этой причине флаксоны, принадлежащие одному классу сопряженных элементов, должны рассматриваться как неразличимые частицы, даже если они предстают во множестве разных лиц (по одному для каждого представителя класса), которые можно различить, выполняя измерения в нужное время, в нужном месте: флаксоны представляют собой *неабелевы анионы*.

8.9. Сплетение неабелевых флаксонов

На примере неабелева сверхпроводника с подходящими свойствами мы увидим, что отказоустойчивый универсальный квантовый компьютер может функционировать, оперируя флаксонами. Самое главное — понять, что происходит при перестановке двух флаксонов.

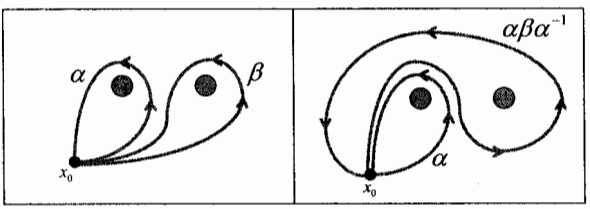
Представим с этой целью, что мы тщательно калибруем два флаксона и сопоставляем им элементы группы G. Это соответствие определяется вы-

бором стандартного базиса для заряженных частиц в начальной точке x_0. Затем выбирается типичный путь, обозначаемый как α, который начинается в точке x_0, обходит против часовой стрелки находящийся *слева* флаксон и возвращается в x_0. Итак, переместив заряженные частицы вдоль замкнутого пути α, мы обнаруживаем, что при этом параллельном переносе их состояния преобразуются матрицей $\mathbf{D}(a)$, где D — представление группы G, по которому преобразуются состояния заряженных частиц, а $a \in G$ — конкретный элемент группы, сопоставляемый флаксону. Аналогично, выбирается другой типичный путь, обозначаемый как β, который начинается в точке x_0, обходит против часовой стрелки флаксон, находящийся *справа*, и возвращается в x_0. Результатом параллельного переноса вдоль β является преобразование состояния заряженной частицы матрицей $\mathbf{D}(b)$, и, таким образом, находящемуся справа флаксону сопоставляется элемент $b \in G$.

Теперь представим, что выполняется перестановка двух флаксонов против часовой стрелки, после чего процедура калибровки повторяется. Какие групповые элементы будут сопоставлены флаксонам теперь?

Чтобы найти ответ, рассмотрим путь $\alpha\beta\alpha^{-1}$; здесь мы используем α^{-1} для обозначения пути α, пройденного в противоположном направлении, и принимаем соглашение, что $\alpha\beta\alpha^{-1}$ обозначает путь, в котором сначала проходится α^{-1}, затем — β и, наконец, α. Теперь видно, что если при перестановке двух флаксонов против часовой стрелки так деформировать пути, чтобы они нигде не пересекались флаксонами, то путь $\alpha\beta\alpha^{-1}$ преобразуется в α, тогда как путь α — в β:

$$\alpha\beta\alpha^{-1} \longmapsto \alpha, \quad \alpha \longmapsto \beta. \tag{8.28}$$



Отсюда следует, что после перестановки результат обхода заряда вдоль пути α эквивалентен результату его перемещения вдоль пути $\alpha\beta\alpha^{-1}$ до перестановки; аналогично, результат обхода вдоль пути β после перестановки эквивалентен результату перемещения вдоль α до перестановки. Мы приходим к выводу, что представляющий перестановку против часовой стрелки оператор сплетения $\mathbf{R}$ действует на флаксоны по правилу

$$\mathbf{R} : |a, b\rangle \longmapsto |aba^{-1}, a\rangle. \tag{8.29}$$

Конечно, если сопоставляемые этим флаксонам a и b являются коммутирующими элементами группы G, то единственным результатом сплетения будет взаимный обмен позициями этих частиц. Но если a и b не коммутируют, результат перестановки более тонкий и интересный. Ассиметрия действия оператора $\mathbf{R}$ является следствием наших соглашений, а также направления перестановки (против часовой стрелки); обратный оператор $\mathbf{R}^{-1}$, представляющий перестановку по часовой стрелке, действует как

$$\mathbf{R}^{-1} : |a, b\rangle \longmapsto |b, b^{-1}ab\rangle. \tag{8.30}$$

Отметим, что суммарный поток пары флаксонов можно детектировать с помощью заряженной частицы, которая обходит путь $\alpha\beta$, окружающий оба элемента данной пары. Поскольку заряд, детектирующий этот суммарный поток, в принципе, может находиться очень и очень далеко, перестановка не должна изменять общий поток; действительно, мы обнаруживаем, что результирующий поток ab сохраняется как оператором $\mathbf{R}$, так и оператором $\mathbf{R}^{-1}$.

Результатом двух последовательных перестановок против часовой стрелки является оператор «монодромии» $\mathbf{R}^2$, представляющий оборот одного флаксона вокруг другого против часовой стрелки и действующий следующим образом:

$$\mathbf{R}^2 : |a, b\rangle \longmapsto |(ab)a(ab)^{-1}, (ab)b(ab)^{-1}\rangle; \tag{8.31}$$

оба потока сопрягаются полным потоком ab, то есть оборот a вокруг b против часовой стрелки ведет к сопряжению b потоком a (и аналогично, оборот b вокруг a по часовой стрелке ведет к сопряжению a потоком b^{-1}). Нетривиальная монодромия означает, что если на плоскости распределено множество флаксонов, один из которых отправляется в мою лабораторию для анализа, то сопосталяемый этому флаксону групповой элемент, вообще говоря, зависит от траектории его перемещения в лабораторию. Если при одном выборе пути квант потока маркируется элементом $a \in G$, то при другом ему, в принципе, может быть сопоставлен любой элемент вида bab^{-1}. Таким образом, инвариантом является представляющий флаксон класс сопряженных элементов в группе G, тогда как конкретный представитель этого класса определяется неоднозначно.

Предположим, например, что G представляет собой S_3 — группу перестановок трех объектов. Один из ее классов сопряженных элементов $\{(12), (23), (31)\}$ имеет порядок три, то есть содержит три элемента — все циклы длины два (перестановки или транспозиции двух объектов —

два-циклы). При объединении (слиянии) пары сопоставляемых этим циклам флаксонов (или, для краткости, два-флаксонов), для суммарного потока существует три возможности — тривиальный квант потока e, или один из квантов потока, сопоставляемых циклам длины три (три-циклам): (123)[1] или (132). Если суммарный поток тривиален, сплетение двух квантов потока также тривиально (a и $b = a^{-1}$ коммутируют). Но если суммарный поток нетривиален, то оператор сплетения $\mathbf{R}$ имеет орбиту длины три:

$$\begin{aligned} &\mathbf{R} : |(12),(23)\rangle \longmapsto |(31),(12)\rangle \longmapsto |(23),(31)\rangle \longmapsto |(12),(23)\rangle, \\ &\mathbf{R} : |(23),(12)\rangle \longmapsto |(31),(23)\rangle \longmapsto |(12),(31)\rangle \longmapsto |(23),(12)\rangle. \end{aligned} \tag{8.32}$$

Таким образом, если два флаксона перестанавливаются трижды, они обмениваются положениями (число перестановок нечетное), но обозначение состояния остается неизменным. Это наблюдение говорит о возможности существования квантовой интерференции между «прямым» и «обменным» рассеянием двух флаксонов, которым сопоставляются разные элементы одного и того же класса сопряженных элементов; это укрепляет представление о том, что флаксоны, переносящие сопряженные «метки», должны рассматриваться как неразличимые частицы.

Поскольку действующий на пары два-флаксонов оператор сплетения удовлетворяет равенству $\mathbf{R}^3 = \mathbf{1}$, его собственные значения равны кубическому корню из единицы. Например, взяв линейные комбинации трех состояний с суммарным потоком (123), мы получим собственные состояния оператора $\mathbf{R}$

$$\begin{aligned} \mathbf{R} = 1: &\quad |(12),(23)\rangle + |(31),(12)\rangle + |(23),(31)\rangle, \\ \mathbf{R} = \omega: &\quad |(12),(23)\rangle + \bar{\omega}|(31),(12)\rangle + \omega|(23),(31)\rangle, \\ \mathbf{R} = \bar{\omega}: &\quad |(12),(23)\rangle + \omega|(31),(12)\rangle + \bar{\omega}|(23),(31)\rangle, \end{aligned} \tag{8.33}$$

где $\omega = e^{2\pi i/3}$, а $\bar{\omega} = \omega^2 = e^{-2\pi i/3}$.

Несмотря на то, что пара флаксонов $|a, a^{-1}\rangle$ с тривиальным суммарным потоком имеет тривиальные свойства сплетения, она интересна по другой причине — она несет *заряд*. Для определения заряда объекта нужно обнести вокруг него квант потока b (против часовой стрелки); это изменяет объект действием матрицы $\mathbf{D}^R(b)$ для некоторого представления R группы G. Если заряд равен нулю, то это представление тривиально: $\mathbf{D}(b) = \mathbf{1}$ для всех значений $b \in G$. Но если мы переносим против часовой стрелки

[1] Напомним, что символ $(n_1 n_2 n_3 \ldots n_k)$ обозначает циклическую перестановку k элементов (k-цикл) $n_1 \to n_2 \to n_3 \to \ldots \to n_k \to n_1$. — *Прим. ред.*

квант потока b вокруг состояния $|a, a^{-1}\rangle$, то это состояние трансформируется в

$$|a, a^{-1}\rangle \longmapsto |bab^{-1}, ba^{-1}b^{-1}\rangle, \tag{8.34}$$

что представляет собой нетривиальное действие (по крайней мере, для некоторых значений b), если a принадлежит классу, содержащему более одного элемента. Действительно, для каждого класса сопряженных элементов α существует единственное состояние $|0;\alpha\rangle$ с нулевым зарядом, однородная суперпозиция представителей класса:

$$|0;\alpha\rangle = \frac{1}{\sqrt{|\alpha|}} \sum_{a\in\alpha} |a, a^{-1}\rangle, \tag{8.35}$$

где $|\alpha|$ означает порядок α. Пара флаксонов класса α, которую можно создать в локальном процессе, не должна нести каких-либо сохраняющихся зарядов и, следовательно, должна находиться в состоянии $|0;\alpha\rangle$. Другие линейные комбинации, ортогональные состоянию $|0,\alpha\rangle$, несут ненулевой заряд. Этот переносимый парой флаксонов заряд можно детектировать с помощью других флаксонов, но, как это ни странно, его нельзя локализовать ни на одной из частиц пары. Скорее, это коллективное свойство пары. При столкновении двух флаксонов с отличным от нуля полным зарядом аннигиляция пары запрещена законом сохранения заряда, даже если их полный поток равен нулю.

Например, для пары флаксонов, соответствующих классу два-циклов группы $G = S_3$, существует двумерное подпространство с тривиальным суммарным потоком и нетривиальным зарядом, в котором можно выбрать базис

$$\begin{aligned} |0\rangle &= |(12),(12)\rangle + \bar{\omega}|(23),(23)\rangle + \omega|(31),(31)\rangle, \\ |1\rangle &= |(12),(12)\rangle + \omega|(23),(23)\rangle + \bar{\omega}|(31),(31)\rangle. \end{aligned} \tag{8.36}$$

При обходе квантом потока b вокруг данной пары, оба ее потока сопрягаются элементом b; следовательно, действие (посредством сопряжения) группы S_3 на эти состояния определяется матрицами

$$\begin{gathered} \mathbf{D}(12) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{D}(23) = \begin{pmatrix} 0 & \bar{\omega} \\ \omega & 0 \end{pmatrix}, \quad \mathbf{D}(31) = \begin{pmatrix} 0 & \omega \\ \bar{\omega} & 0 \end{pmatrix}, \\ \mathbf{D}(123) = \begin{pmatrix} \omega & 0 \\ 0 & \bar{\omega} \end{pmatrix}, \quad \mathbf{D}(132) = \begin{pmatrix} \bar{\omega} & 0 \\ 0 & \omega \end{pmatrix}, \end{gathered} \tag{8.37}$$

образующими ее двумерное неприводимое представление $R = [2]$. Таким образом, мы приходим к выводу, что заряд этой пары флаксонов равен $[2]$.

Более того, при сплетении этот переносимый парой флаксонов заряд может перейти на другие частицы. Рассмотрим, например, пару частиц, каждая из которых несет заряд, но не имеет кванта потока (я буду называть такие частицы *чарджионами*), так что полный заряд пары тривиален. Если состояние одного из чарджионов преобразуется по унитарному неприводимому представлению R группы G, то существует единственное сопряженное представление $\bar{R}$, которое можно скомбинировать с R, чтобы получить тривиальное представление; если $\{|R,i\rangle\}$ является базисом для R, то для $\bar{R}$ можно выбрать такой базис $\{|\bar{R},i\rangle\}$, в котором состояние пары чарджионов с тривиальным зарядом представляется в виде

$$|0;R\rangle = \frac{1}{\sqrt{|R|}} \sum_i |R,i\rangle \otimes |\bar{R},i\rangle. \tag{8.38}$$

Предположим, что мы создали пару флаксонов в состоянии $|0;\alpha\rangle$ и пару чарджионов в состоянии $|0;R\rangle$. Затем обводим против часовой стрелки чарджион с зарядом R вокруг флаксона с квантом потока класса α и снова сталкиваем два чарджиона, чтобы посмотреть, не аннигилируют ли они. Что же происходит?

При фиксированном значении кванта потока $a \in \alpha$ влияние обхода на состояние пары чарджионов согласно (8.27) состоит в следующем:

$$|0;R\rangle \longmapsto \frac{1}{\sqrt{|R|}} \sum_{i,j} |R,j\rangle \otimes |\bar{R},i\rangle D^R_{ji}(a); \tag{8.39}$$

если сейчас измерить заряд пары, то вероятность получения нулевого полного заряда равна квадрату модуля перекрытия этого состояния с $|0;R\rangle$, то есть

$$\mathrm{Prob}(0) = \left| \frac{\chi^R(a)}{|R|} \right|^2, \tag{8.40}$$

где

$$\chi^R(a) = \sum_i D^R_{ii}(a) = \mathrm{tr}\mathbf{D}^R(a), \tag{8.41}$$

— *характер* представления R, вычисленный для данного элемента a. Действительно, характер (след) инвариантен относительно операции сопряжения — он принимает одно и то же значение для всех $a \in \alpha$. Следовательно, (8.40) также представляет собой вероятность того, что полный заряд пары чарджионов останется равным нулю, если один из чарджионов (элемент соответствующей пары в исходном состоянии $|0;R\rangle$) обойдет вокруг

одного из флаксонов (элемента соответствующей пары в состоянии $|0;\alpha\rangle$). Конечно, поскольку полный заряд всех четырех частиц равен нулю, а заряд сохраняется, то после обхода эти две пары приобретают противоположные заряды: если пара чарджионов имеет заряд R', то пара флаксонов должна приобрести заряд $\bar{R}'$, который, комбинируясь с R', дает в результате тривиальный полный заряд. Пара частиц с нулевым полным зарядом и квантом потока может аннигилировать, не оставляя после себя ни одной стабильной частицы, тогда как пара с ненулевым зарядом полностью аннигилировать не может. Таким образом, если мировые линии пар флаксонов и чарджионов сцепляются один раз, вероятность того, что обе пары будут способны аннигилировать, задается уравнением (8.40). Эта вероятность меньше единицы, при условии, что представление R не одномерно, а класс α представлен нетривиально. Таким образом, сцепление мировых линий индуцирует обмен зарядами между двумя парами.

Например, в случае, когда α является классом два-циклов группы $G = S_3$, а $R = [2]$ (двумерное неприводимое представление группы S_3), из уравнений (8.37) следует, что $\chi^{[2]}(\alpha) = 0$. Таким образом, в этом случае вероятность обмена зарядом равна единице; после оборота обе пары флаксонов и чарджионов преобразуются по $R' = [2]$.

8.10. Суперотборные секторы неабелева сверхпроводника

В предыдущем обсуждении неабелева сверхпроводника рассматривались два типа частиц: *флаксоны*, несущие квант потока, но не имеющие заряда, и *чарджионы*, несущие заряд, но не поток. Это не самые общие возможные частицы. Поучительно рассмотреть, что получится, если создать сложную частицу, составленную из флаксона и чарджиона. В частности, чему будет равен заряд такого композита? Это на редкость тонкий вопрос; чтобы дать обоснованный ответ, нужно как следует подумать о том, как можно измерить заряд.

В принципе, заряд можно измерить с помощью интерференционного эксперимента Аронова–Бома. Объект, заряд которого мы хотим определить, можно поместить непосредственно за экраном между двумя (прорезанными в нем параллельными) щелями, направить на этот экран пучок тщательно откалиброванных флаксонов и детектировать их по ту сторону экрана. Выявленные таким образом сдвиг и контрастность интерференционной картины позволяют определить $\mathbf{D}^R(b)$ для каждого $b \in G$ и таким образом определить R.

Однако все не так просто, если вместе с зарядом анализируемый объект несет нетривиальный поток $a \in G$. Поскольку перенос потока b во-

круг a меняет a на bab^{-1}, то в случае некоммутирующих a и b два возможных пути для потока b не интерферируют. Действительно, после детектирования потока b можно проверить, не изменился ли поток a, и, следовательно, определить, прошел поток b сквозь щель слева или справа от a. Поскольку поток (a или bab^{-1}) коррелирует с информацией «какой путь» (левая или правая щель), интерференция разрушается.

Следовательно, этот эксперимент раскрывает информацию о заряде, только при коммутирующих a и b. Поэтому прикрепленный к потоку a заряд не описывается неприводимым представлением группы G; вместо этого он описывается как неприводимое представление подгруппы группы G, нормализатора $N(a)$ потока a в группе G, который определяется как

$$N(a) = \{b \in G | ab = ba\}. \tag{8.42}$$

Нормализаторы $N(a)$ и $N(bab^{-1})$ изоморфны, поэтому нормализатор естественнее ассоциировать с классом α группы G, а не с конкретным ее элементом, и обозначать как $N(\alpha)$. Следовательно, каждый тип частиц, возникающих в нашем неабелевом сверхпроводнике, в действительности имеет две метки: класс сопряженных элементов α, характеризующий поток, и описывающее заряд неприводимое представление $R^{(\alpha)}$ нормализатора $N(\alpha)$. Принято говорить, что α и $R^{(\alpha)}$ характеризуют *суперотборные секторы* теории, поскольку они представляют свойства локализованного объекта, которые обязаны сохраняться в любых локальных физических процессах. Для частиц, несущих метки $(\alpha, R^{(\alpha)})$, можно учредить «бюро стандартов», обращаясь к которому в конкретное время и в конкретном месте, можно распознать всего $|\alpha| \cdot |R^{(\alpha)}| \equiv d_{(\alpha, R(\alpha))}$ различных видов частиц — это число называется *размерностью* сектора. Если эти частицы сплетаются с другими, их виды могут меняться, тогда как метки $(\alpha, R^{(\alpha)})$ остаются неизменными.

В любой теории анионов каждому типу частиц можно приписать размерность, хотя, как мы увидим, в общем случае она не обязана быть целой и толковаться как количество различных видов частиц одного типа (одного сектора). *Полную размерность* $\mathcal{D}$ можно определить, просуммировав по всем типам; в случае неабелева сверхпроводника мы имеем

$$\mathcal{D}^2 = \sum_{\alpha} \sum_{R^{(\alpha)}} d^2_{(\alpha, R^{(\alpha)})} = \sum_{\alpha} |\alpha|^2 \sum_{R^{(\alpha)}} |R^{(\alpha)}|^2. \tag{8.43}$$

Поскольку сумма квадратов размерностей всех неприводимых представлений конечной группы равна ее порядку, а порядок нормализатора $N(\alpha)$ ра-

вен $|G|/|\alpha|$, мы получаем

$$\mathcal{D}^2 = \sum_{\alpha} |\alpha| \cdot |G| = |G|^2, \tag{8.44}$$

а полная размерность $\mathcal{D} = |G|$.

В случае $G = S_3$ существует восемь типов частиц, которые перечислены ниже:

Тип	Поток	Заряд	Размерность
A	e	$[+]$	1
B	e	$[-]$	1
C	e	[2]	2
D	(12)	$[+]$	3
E	(12)	$[-]$	3
F	(123)	[1]	2
G	(123)	$[\omega]$	2
H	(123)	$[\bar{\omega}]$	2

Если поток тривиален (e), тогда заряд может быть одним из трех неприводимых представлений группы S_3: тривиальным одномерным представлением $[+]$, нетривиальным одномерным представлением $[-]$ или двумерным представлением [2]. Если поток представляет собой два-цикл, то нормализатором является Z_2, а зарядом может быть либо тривиальное представление $[+]$, либо нетривиальное $[-]$. Если же потоком является 3-цикл, тогда нормализатором будет Z_3, а заряд может быть либо тривиальным представлением [1], либо нетривиальным представлением $[\omega]$, либо сопряженным ему представлением $[\bar{\omega}]$. Вы можете проверить, что полная размерность, как и ожидалось, равна $\mathcal{D} = |S_3| = 6$.

Отметим, что, поскольку a по определению коммутирует со всеми элементами нормализатора $N(a)$, матрица $\mathbf{D}^{R^{(a)}}(a)$, представляющая a в неприводимом представлении $R^{(a)}$, коммутирует со всеми матрицами в этом представлении; следовательно, согласно лемме Шура она кратна единице:

$$\mathbf{D}^{R^{(a)}}(a) = \exp(i\theta_{R^{(a)}})\mathbf{1}. \tag{8.45}$$

Чтобы понять смысл фазы $\exp(i\theta_{R^{(a)}})$, рассмотрим композит поток–заряд, в котором чарджион в представлении $R^{(a)}$ привязан к потоку a, и представим поворот против часовой стрелки этого композитного объекта на 2π.

Этот поворот переносит заряд вокруг потока, генерируя фазу

$$e^{-2\pi i\mathbf{J}} = e^{i\theta_{R^{(a)}}}; \tag{8.46}$$

следовательно, каждому суперотборному сектору соответствует определенное значение *топологического спина*, определяемое как $\theta_{R^{(a)}}$.

При слиянии двух разных типов частиц могут получиться композитные объекты различных типов, а какие типы возможны, устанавливают *правила слияния* данной теории. Квант потока композита может принадлежать любому из классов сопряженных элементов, которые можно получить перемножением представителей классов, соответствующих двум составляющим рассматриваемого объекта. Определение заряда композита особенно сложно, поскольку для этого необходимо разложить тензорное произведение представлений двух разных нормализаторов на сумму представлений нормализатора результирующего потока. Например, в случае $G = S_3$ правило, управляющее слиянием двух частиц типа D, выглядит следующим образом:

$$\mathrm{D} \times \mathrm{D} = \mathrm{A} + \mathrm{C} + \mathrm{F} + \mathrm{G} + \mathrm{H}. \tag{8.47}$$

Мы уже отмечали, что слияние пары потоков, сопоставляемых два-циклам, может дать либо тривиальный суммарный поток, либо поток соответствующий три-циклу. Заряд композита с тривиальным суммарным потоком может быть либо [+], либо [2]. Если же суммарный поток, соответствует три-циклу, то собственные зарядовые состояния в точности совпадают с собственными состояниями оператора сплетения, построенными в уравнении (8.33).

Почему собственные состояния полного заряда системы двух анионов должны также являться и собственными состояниями оператора сплетения? Мы можем понять эту связь в более общем смысле, если подумаем об угловом моменте двух-анионного составного объекта. Оператор монодромии $\mathbf{R}^2$ заключает в себе результат обхода против часовой стрелки одной частицы вокруг другой. Этот обход — практически то же самое, что и поворот против часовой стрелки композитной системы на угол 2π, за исключением того, что при повороте композитной системы вращаются и обе ее составляющие. Вращение этих составляющих будет компенсироваться, если поворот композита против часовой стрелки будет сопровождаться поворотом его составляющих по часовой стрелке. Следовательно, оператор монодромии можно представить в виде

$$(\mathbf{R}^c_{ab})^2 = e^{-2\pi i\mathbf{J}_c} e^{2\pi i\mathbf{J}_a} e^{2\pi i\mathbf{J}_b} = e^{i(\theta_c - \theta_a - \theta_b)}, \tag{8.48}$$

где $\mathbf{R}^c_{ab}$ обозначает оператор сплетения для перестановки против часовой стрелки частиц типа a и b, соединеных в композит типа c. Здесь мы поль-

зуемся более коротким, чем раньше, обозначением, в котором a, b, c представляют набор меток для суперотборных секторов (определяющих в модели неабелева сверхпроводника и поток, и заряд). Поскольку каждый суперотборный сектор имеет определенный топологический спин, а оператор монодромии диагонален в базисе топологического спина, мы видим, что собственные состояния заряда и оператора сплетения совпадают. Отметим, что уравнение (8.48) обобщает наши предшествующие наблюдения относительно абелевых анионов, а именно: композит из двух идентичных анионов имеет топологический спин $e^{i4\theta}$, а обменная фаза анион-антианионнной пары (с тривиальным полным спином) равна $e^{-i\theta}$.

8.11. Квантовые вычисления с неабелевыми флаксонами

Модель анионов характеризуется ответами на два основных вопроса: (1) Что происходит при комбинировании двух анионов (в чем состоят *правила слияния*)? (2) Что происходит при перестановке двух анионов (в чем состоят *правила сплетения*)? Мы обсудили ответы на эти вопросы в частном случае модели неабелева сверхпроводника, ассоциируемой с конечной неабелевой группой G, а сейчас мы хотели бы понять, как эти правила слияния и сплетения можно применить для моделирования квантовых схем.

Описывая это моделирование, мы будем предполагать следующие физические возможности:

Рождение и идентификация пары. Мы можем создавать пары частиц и определять тип частиц каждой пары [класс сопряженных элементов α потока каждой частицы в паре, а также заряд частицы — неприводимое представление $R^{(\alpha)}$ нормализатора группы потока $N(\alpha)$]. Это предположение естественно, поскольку не существует симметрии, связывающей частицы разных типов; они имеют различимые физические свойства — например, различные энергетические щели и эффективные массы. В самом деле, единственные типы частиц, которые нам потребуются, — это не имеющие заряда флаксоны и не несущие потока чарджионы.

Аннигиляция пары. Мы можем сталкивать две частицы друг с другом и наблюдать, аннигилирует ли эта пара полностью. Таким образом, мы получаем ответ на вопрос: имеет эта пара частиц тривиальный поток и заряд или нет? Это предположение естественно, поскольку если пара имеет нетривиальное значение какой-либо сохраняющейся величины, то после ее слияния должно остаться локализованное возбуждение, а эта остаточная частица, в принципе, обнаружима.

Сплетение. Мы можем вести частицы вдоль установленных траекторий и, таким образом, выполнять их перестановки. Квантовые вентили будут моделироваться выбором мировых линий частиц, реализующих конкретные косы.

Эти элементарные операции позволяют реализовать некоторые выводимые ниже возможности, которыми мы будем регулярно пользоваться. Во-первых, мы можем использовать чарджионы для калибровки флаксонов и учредить бюро стандартов для потоков. Предположим, нам вручили две пары флаксонов в состояниях $|a, a^{-1}\rangle$ и $|b, b^{-1}\rangle$. Мы хотим сравнить потоки a и b. Для этого создадим пару чарджион-античарджион, где заряд чарджиона представляет собой неприводимое представление R группы G. Затем пронесем чарджион вдоль замкнутого пути, окружающего первый элемент первой пары флаксонов и второй элемент второй пары флаксонов. И наконец, вновь соединим чарджион с античарджионом и посмотрим, аннигилируют они или нет. Поскольку ограниченный траекторией чарджиона суммарный поток равен ab^{-1}, пара чарджион-античарджион аннигилирует с вероятностью

$$\text{Prob}(0) = \left| \frac{\chi^R(ab^{-1})}{|R|} \right|^2, \tag{8.49}$$

что меньше единицы, если поток ab^{-1} не равен единице (при условии, что представление R не одномерное и представляет ab^{-1} нетривиально). Таким образом, если аннигиляция не происходит, мы знаем наверняка, что потоки a и b не совпадают; если же чарджионные пары всякий раз аннигилируют, то равенство a и b становится все более вероятным. Повторив эту процедуру умеренное количество раз, можно с высокой статистической достоверностью сделать вывод о равенстве a и b.

Эта процедура позволяет рассортировать пары флаксонов по корзинам так, чтобы все пары, лежащие в одной корзине, имели одинаковый квант потока. Если корзина содержит n пар, то в общем случае ее состояние представляет собой смесь состояний вида

$$\sum_{a \in G} \psi_a |a, a^{-1}\rangle^{\otimes n}. \tag{8.50}$$

После удаления из корзины лишь одной пары каждое такое состояние становится смесью

$$\sum_{a \in G} \rho_a (|a, a^{-1}\rangle\langle a, a^{-1}|)^{\otimes (n-1)}; \tag{8.51}$$

каждую корзину можно рассматривать как содержащую $(n-1)$ пару с одним и тем же определенным, но пока неизвестным потоком.

«Кто есть кто» в этих корзинах? Мы хотим маркировать корзины элементами группы G. Чтобы прийти к согласованной маркировке, извлечем по паре флаксонов из трех разных корзин. Предположим, что эти три пары имеют вид $|a, a^{-1}\rangle$, $|b, b^{-1}\rangle$ и $|c, c^{-1}\rangle$, и попытаемся проверить, справедливо ли равенство $c = ab$. Для этого создадим пару чарджион-античарджион и пронесем чарджион вдоль замкнутого пути, окружающего первый элемент первой пары флаксонов, первый элемент второй пары флаксонов и второй элемент третьей пары флаксонов, после чего посмотрим, аннигилирует вновь объединенная чарджионная пара или нет. Поскольку суммарный поток, окруженный траекторией чарджиона равен abc^{-1}, повторяя эту процедуру, можно с высокой статистической уверенностью определить, равны ли ab и c. Эти наблюдения позволяют маркировать корзины в соответствии с групповым правилом композиции. Такая маркировка является однозначной, за исключением групповых автоморфизмов (а неоднозначности, возникающие от любых автоморфизмов, можно разрешить произвольным образом).

Как только бюро стандартов для потоков создано, его можно использовать для измерения неизвестного потока немаркированной пары. Если состояние измеряемой пары — $|d, d^{-1}\rangle$, мы можем извлечь из корзины маркированную пару $|a, a^{-1}\rangle$ и использовать чарджионные пары для измерения потока ad^{-1}. Повторяя эту процедуру с другими маркированными потоками и, в конечном счете, осуществляя проецирующее измерение потока, мы можем определить значение d.

Для моделирования квантовых схем с помощью флаксонов нам потребуется выполнять логические вентили, действующие на значение потока. Основной вентиль, который мы будем использовать, реализуется путем оборота против часовой стрелки пары флаксонов в состоянии $|a, a^{-1}\rangle$ вокруг первого элемента другой пары флаксонов в состоянии $|b, b^{-1}\rangle$. Поскольку пара $|a, a^{-1}\rangle$ имеет тривиальный полный поток, данная процедура никак не влияет на пару $|b, b^{-1}\rangle$. Но поскольку в результате поток b обходит против часовой стрелки вокруг обоих элементов пары, начальным состоянием которой было $|a, a^{-1}\rangle$, то эта пара преобразуется как

$$|a, a^{-1}\rangle \longmapsto |bab^{-1}, ba^{-1}b^{-1}\rangle. \tag{8.52}$$

Будем называть это преобразование *операцией сопряжения*, действующей на пару флаксонов.

Подведем итог вышесказанному. Наши исходные и выведенные возможности позволяют: (1) выполнять проецирующее измерение потока,

(2) выполнять разрушающее измерение, определяющее, являются ли поток и заряд пары тривиальными и (3) осуществлять сопрягающий элемент. Сейчас мы должны обсудить, как моделировать квантовые схемы, используя эти возможности.

Нужно решить, как кодировать кубиты с помощью флаксонов. Подходящее кодирование можно выбрать разными способами; мы остановимся на одном конкретном выборе, иллюстрирующем ключевые идеи, а именно: будем кодировать кубит, используя пару флаксонов, полный поток которой тривиален. Выберем два некоммутирующих элемента $a, b \in G$, где $b^2 = e$, и вычислительный базис для кубита

$$|\bar{0}\rangle = |a, a^{-1}\rangle, \quad |\bar{1}\rangle = |bab^{-1}, ba^{-1}b^{-1}\rangle. \tag{8.53}$$

Важный нюанс: отдельный изолированный флаксон с квантом потока a выглядит идентичным флаксону с сопряженным квантом потока bab^{-1}. Следовательно, если два флаксона из одной пары удерживать далеко друг от друга, локальные взаимодействия с окружающей средой не вызовут декогерентизации суперпозиции состояний $|\bar{0}\rangle$ и $|\bar{1}\rangle$. Квантовая информация защищена от повреждения, поскольку она хранится нелокальным образом, благодаря топологическому вырождению состояний, в которых флаксон и антифлаксон закреплены в фиксированных пространственно разделенных позициях.

Но в отличие от топологического вырождения в системах с абелевыми анионами, этот защищенный кубит можно сравнительно просто измерить, не прибегая к тонким интерферометрическим процедурам, выделяющим фазы Ааронова–Бома. Мы уже описывали, как измерить поток, используя предварительно откалиброванные флаксоны; следовательно, мы можем выполнить проецирующее измерение закодированного оператора Паули $\bar{\mathbf{Z}}$ (проекцию на базис $\{|\bar{0}\rangle, |\bar{1}\rangle\}$). Мы также можем измерить дополнительный оператор Паули $\bar{\mathbf{X}}$, пусть даже разрушающим методом и неидеально. Собственными состояниями оператора $\bar{\mathbf{X}}$ являются

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|\bar{0}\rangle \pm |\bar{1}\rangle) \equiv \frac{1}{\sqrt{2}}(|a, a^{-1}\rangle \pm |bab^{-1}, ba^{-1}b^{-1}\rangle); \tag{8.54}$$

следовательно, состояние $|-\rangle$ *ортогонально* состоянию нулевого заряда

$$|0; \alpha\rangle = \frac{1}{\sqrt{|\alpha|}} \left(\sum_{c \in \alpha} |c, c^{-1}\rangle \right), \tag{8.55}$$

где α — класс сопряженных элементов, содержащий a. С другой стороны, состояние $|+\rangle$ имеет ненулевое перекрытие с $|0;\alpha\rangle$

$$\langle +|0;\alpha\rangle = \sqrt{2/|\alpha|}. \tag{8.56}$$

Следовательно, при столкновении частиц, образующих флаксонную пару, полная аннигиляция невозможна, если пара находится в состоянии $|-\rangle$, если же состоянием пары является $|+\rangle$, то аннигиляция происходит с вероятностью $\mathrm{Prob}(0) = 2/|\alpha|$.

Отметим, что можно также приготовить флаксонную пару в состоянии $|+\rangle$. Один из возможных способов состоит в следующем: сначала создается пара в состоянии $|0;\alpha\rangle$. Если α содержит только два элемента a и bab^{-1}, то задача решена. В противном случае вновь созданная пара сравнивается с калиброванными парами в каждом из состояний $|c, c^{-1}\rangle$, где $c \in \alpha$ и отличается как от a, так и от bab^{-1}. Если она не согласуется ни с одной из этих $|c, c^{-1}\rangle$-пар, то ее состоянием должно быть $|+\rangle$.

Чтобы продвинуться дальше, необходимо определить вычислительные возможности операции сопряжения. Будем использовать более компактное обозначение, в котором состояние $|x, x^{-1}\rangle$ флаксонной пары обозначается просто как $|x\rangle$, и рассмотрим преобразования состояния $|x, y, z\rangle$, которые можно построить с помощью операции сопряжения. Пронося (по замкнутому пути) третью пару сквозь первую против или по часовой стрелке, можно реализовать следующие вентили:

$$|x, y, z\rangle \longmapsto |x, y, xzx^{-1}\rangle, \quad |x, y, z\rangle \longmapsto |x, y, x^{-1}zx\rangle. \tag{8.57}$$

Точно так же, пронося третью пару сквозь вторую против или по часовой стрелки, можно реализовать вентили

$$|x, y, z\rangle \longmapsto |x, y, yzy^{-1}\rangle, \quad |x, y, z\rangle \longmapsto |x, y, y^{-1}zy\rangle. \tag{8.58}$$

Более того, позаимствовав в бюро стандартов пару с квантом потока $|c\rangle$, можно осуществить

$$|x, y, z\rangle \longmapsto |x, y, czc^{-1}\rangle \tag{8.59}$$

для любой постоянной величины $c \in G$. Комбинация этих элементарных операций позволяет реализовать любой вентиль вида

$$|x, y, z\rangle \longmapsto |x, y, fzf^{-1}\rangle, \tag{8.60}$$

где функцию $f(x, y)$ можно представить в *мультипликативной форме*, то есть в виде конечного произведения, сомножителями которого могут быть

входящие данные x и y, обратные им значения x^{-1} и y^{-1} или постоянные элементы группы G, причем каждый из них может встречаться в произведении любое количество раз.

Что представляют собой функции $f(x,y)$, которые можно представить в таком виде? Ответ зависит от структуры группы G, но для наших целей достаточно следующей характеристики. Вспомним, что подгруппа H конечной группы G является *нормальной*, если для любой величины $h \in H$ и $g \in G$, $ghg^{-1} \in H$;[1] вспомним также, что конечная группа G является *простой*, если G не имеет нормальных подгрупп за исключением самой группы G и тривиальной подгруппы $\{e\}$. Оказывается, если G является простой неабелевой конечной группой, то *любую* функцию $f(x,y)$ можно представить в мультипликативной форме. В литературе по теории вычислительных систем родственный этому утверждению результат часто называют *теоремой Баррингтона*.

В частности, если группа G представляет собой неабелеву простую группу, то существует представимая в мультипликативной форме функция f, такая, что

$$f(a,a) = f(a,bab^{-1}) = f(bab^{-1},a) = e, \quad f(bab^{-1},bab^{-1}) = b. \qquad (8.61)$$

Таким образом, для $x,y,z \in \{a,bab^{-1}\}$ действие уравнения (8.60) «переворачивает» поток третьей пары, если и только если $x = y = bab^{-1}$; из наших элементарных операций мы построили вентиль Тоффоли в вычислительном базисе. Следовательно, для реализации универсальных обратимых *классических* вычислений достаточно операций сопряжения, действующих на стандартные базисные состояния.

A_5 — неабелева простая группа минимального порядка $|A_5| = 60$, группа четных перестановок пяти объектов. Следовательно, одна конкретная реализация универсальных классических вычислений, использующих операции сопряжения, получается при выборе в качестве a три-цикла элемента $a = (345) \in A_5$, а в качестве b — произведения пары два-циклов $b = (12)(34) \in A_5$, так что $bab^{-1} = (435)$.

При таком разумном выборе группы G мы добиваемся топологической реализации универсальных классических вычислений, но как продвинуться еще дальше и осуществить универсальные квантовые вычисления? Мы имеем возможность готовить состояния вычислительного базиса, измерять в вычислительном базисе и выполнять вентили Тоффоли, но все это чисто классические средства. Единственными неклассическими приемами, которыми мы располагаем, являются способность готовить собственные состо-

[1] Такие подгруппы называют также *инвариантными*, или *нормальными, делителями*.

яния $\bar{\mathbf{X}} = 1$ и умение выполнять неидеальное разрушающее измерение $\bar{\mathbf{X}}$. К счастью, этих дополнительных возможностей оказывается достаточно.

В предыдущем обсуждении квантовой отказоустойчивости мы отмечали, что если можно выполнять классические вентили Тоффоли и CNOT, то для универсальных квантовых вычислений достаточно уметь применять каждый из операторов Паули $\mathbf{X}$, $\mathbf{Y}$ и $\mathbf{Z}$, а также выполнять проецирующие измерения любого из этих операторов. Мы уже знаем, как применить классический вентиль $\mathbf{X}$ и как измерить $\mathbf{Z}$ (то есть спроецировать на вычислительный базис). В нашем арсенале до сих пор отсутствуют проецирующие измерения $\mathbf{X}$ и $\mathbf{Y}$ и выполнение $\mathbf{Z}$. (Конечно, если мы можем применять операторы $\mathbf{X}$ и $\mathbf{Z}$, то в состоянии применять и их произведение $\mathbf{ZX} = i\mathbf{Y}$.)

Посмотрим, как превратить неидеальное разрушающее измерение $\mathbf{X}$ в надежное проецирующее измерение. Вспомним действие сопряжения CNOT на операторы Паули:

$$\text{CNOT} : \mathbf{X1} \longmapsto \mathbf{XX}, \tag{8.62}$$

где первый кубит является управляющим, а второй — целью CNOT. Следовательно, для осуществления проецирующего измерения $\mathbf{X}$ достаточно вентилей CNOT, а также способности готовить собственные состояния $\mathbf{X} = 1$ и выполнять разрушающие измерения $\mathbf{X}$. Мы можем приготовить служебный кубит в собственном состоянии $\mathbf{X} = 1$, выполнить вентиль CNOT со служебным кубитом в качестве управляющего и измеряемыми данными в качестве цели, а затем провести разрушающее измерение этого служебного кубита. Это измерение готовит данные в собственном состоянии оператора $\mathbf{X}$, собственное значение которого совпадает с результатом измерения служебного кубита. В нашем случае разрушающее измерение не вполне надежно, но его можно многократно повторить. Каждый раз мы готовим и измеряем новый служебный бит и после нескольких повторений получаем результат измерения с приемлемой статистической достоверностью.

Теперь, когда мы можем выполнить проецирующее измерение $\mathbf{X}$, мы способны приготавливать не только собственные состояния $\mathbf{X} = 1$, но и собственные состояния $\mathbf{X} = -1$ (например, повторяя следующие друг за другом измерения операторов $\mathbf{Z}$ и $\mathbf{X}$ до тех пор, пока в конце концов не будет получен результат $\mathbf{X} = -1$). Затем, выполняя вентиль CNOT, целью которого является собственное состояние $\mathbf{X} = -1$, мы можем выполнить оператор Паули $\mathbf{Z}$, действующий на управляющий кубит. Остается лишь продемонстрировать, что измерение $\mathbf{Y}$ тоже может быть реализовано.

На первый взгляд, измерение оператора $\mathbf{Y}$ выглядит проблематичным, поскольку наши физические возможности не позволяют различать собственные состояния $\mathbf{Y} = 1$ и $\mathbf{Y} = -1$ (то есть отличать состояние ψ от

его комплексно-сопряженного ψ^*). Но эта неопределенность не представляет серьезной трудности, поскольку на самом деле не важно, как она будет разрешена. Если в процедуре моделирования унитарного преобразования $\mathbf{U}$ заменить измерение оператора $\mathbf{Y}$ на измерение оператора $-\mathbf{Y}$, то результатом такой замены будет моделирование $\mathbf{U}^*$ вместо $\mathbf{U}$; эта замена не меняет распределение вероятностей результатов измерений в стандартном вычислительном базисе.

Чтобы быть точнее, можно сформулировать протокол измерения оператора $\mathbf{Y}$, в первую очередь отметив, что применение вентиля Тоффоли, целевым кубитом которого является собственное состояние $\mathbf{X} = -1$, реализует вентиль контролируемой фазы $\Lambda(\mathbf{Z})$, действующий на два управляющих кубита. Объединив этот вентиль с CNOT вентилем $\Lambda(\mathbf{X})$, мы получаем вентиль $\Lambda(i\mathbf{Y})$, действующий как

$$\Lambda(i\mathbf{Y}): \quad \begin{cases} |X=+1\rangle \otimes |Y=+1\rangle \longmapsto |Y=+1\rangle \otimes |Y=+1\rangle, \\ |X=+1\rangle \otimes |Y=-1\rangle \longmapsto |Y=-1\rangle \otimes |Y=-1\rangle, \\ |X=-1\rangle \otimes |Y=+1\rangle \longmapsto |Y=-1\rangle \otimes |Y=+1\rangle, \\ |X=-1\rangle \otimes |Y=-1\rangle \longmapsto |Y=+1\rangle \otimes |Y=-1\rangle, \end{cases} \tag{8.63}$$

где первый кубит является управляющим, а второй — целью. Теперь предположим, что мой надежный друг дает мне только один кубит, который, как он меня уверяет, был приготовлен в состоянии $|Y=1\rangle$. Я знаю, как самостоятельно приготовить состояния $|X=1\rangle$, и могу выполнять вентили $\Lambda(i\mathbf{Y})$; следовательно, поскольку вентиль $\Lambda(i\mathbf{Y})$ с состоянием $|Y=1\rangle$ в качестве его цели преобразует $|X=1\rangle$ в $|Y=1\rangle$, я могу сделать множество копий полученного от моего друга состояния $|Y=1\rangle$. Желая измерить $\mathbf{Y}$, я применяю обратный по отношению к $\Lambda(i\mathbf{Y})$ вентиль, целью которого является измеряемый кубит, а управляющим кубитом — одно из моих состояний $\mathbf{Y}=1$; затем я выполняю $\mathbf{X}$-измерение служебного кубита, считывающие результат $\mathbf{Y}$-измерения другого кубита.

А что, если мой друг солгал и вместо этого дал мне копию состояния $|Y=-1\rangle$? Тогда я сделаю множество копий состояния $|Y=-1\rangle$ и буду измерять $-\mathbf{Y}$, считая, что измеряю $\mathbf{Y}$. Мое моделирование будет работать точно так же, как и раньше; фактически я буду моделировать комплексно-сопряженное значение идеальной схемы, но это не изменит конечного результата квантового вычисления. Если мой друг подбросит в воздух монетку, чтобы решить, дать ли мне состояние $|Y=1\rangle$ или $|Y=-1\rangle$, это также не окажет никакого влияния на точность моего моделирования. То есть получается, что помощь друга мне вообще не нужна — вместо использования состояния $|Y=1\rangle$, которое я мог бы получить от него, я могу

использовать случайное состояние $\rho = 1/2$ (равновзвешенную смесь состояний $|Y = 1\rangle$ и $|Y = -1\rangle$), способ приготовления которого мне известен.

Этим завершается доказательство того, что по крайней мере в случае, когда G является простой неабелевой конечной группой,[1] мы можем эффективно моделировать отказоустойчивые квантовые схемы, используя флаксоны и чарджионы неабелева сверхпроводника. Рассматриваемое в целом, включая все приготовления состояний и калибровку потоков, это моделирование можно описать следующим образом: приготавливается множество пар анионов (флаксонов и чарджионов), мировые линии анионов следуют определенной косе, анионные пары сталкиваются с целью увидеть, аннигилируют они или нет. Это моделирование недетерминированно в том смысле, что реализованная анионами фактическая коса зависит от результатов измерений, выполненных (благодаря столкновениям) в процессе моделирования. Оно устойчиво к малым возмущениям, если температура низка по сравнению с энергетической щелью, а частицы удерживаются достаточно далеко друг от друга (за исключением момента рождения и столкновения пар), чтобы подавить обмен виртуальными анионами. Пока сплетение частиц принадлежит правильному топологическому классу, малые деформации их мировых линий не влияют на результат вычислений.

8.12. Обобщенные анионные модели

Наше обсуждение модели неабелева сверхпроводника обеспечивает доказательство существования использующих анионы отказоустойчивых квантовых вычислений. Но эта модель, естественно, имеет недостатки. Описанной нами схеме недостает красоты, элегантности, простоты.

Я рассмотрел эту модель так подробно, поскольку она достаточно конкретна и, следовательно, позволяет нам приобрести интуицию в понимании свойств неабелевых анионов. Но сейчас, когда мы уже лучше понимаем ключевые идеи сплетения и слияния в анионных моделях, мы готовы подойти к этим вопросам с более общих и абстрактных позиций. Это приведет нас к новым моделям, в том числе и к существенно более простым по сравнению рассмотренными ранее. Мы сможем выбросить большую часть лишнего багажа, обременяющего модель неабелева сверхпроводника (например, различие между флаксонами и чарджионами, калибровку флаксонов, а также измерения, требуемые для моделирования неклассических

[1] Мошон показал, что универсальные квантовые вычисления возможны для более широкого класса групп. См.: C. Mochon, *Anyons from Non-Solvable Finite Groups are Sufficient for Universal Quantum Computation*, Phys. Rev. **A67**, 022315 (2003).

вентилей). Простейшие модели, с которыми мы сейчас познакомимся, органичнее вписываются в схемы отказоустойчивых вычислений и выглядят более правдоподобными с точки зрения их возможной реализации в системах с разумными физическими свойствами.

Анионная модель — это теория частиц на двумерной поверхности (которую мы будем считать плоской), переносящих локально сохраняющиеся *заряды*. Также предполагается, что эта теория имеет массовую щель, то есть дальнодействующие взаимодействия, переносимые безмассовыми частицами, отсутствуют. Эта модель характеризуется тремя определяющими свойствами:

1. Перечень *типов* частиц. Типами являются метки, устанавливающие возможные значения сохраняющегося заряда, который может нести частица.

2. Правила *композиции* и *расщепления*, определяющие возможные значения заряда, которые могут быть получены при объединении двух частиц с известными зарядами, а также возможные способы расщепления на две части заряда, переносимого одной частицей.

3. Правила *сплетения*, устанавливающие, что происходит при перестановке двух частиц (или при повороте одной частицы на 2π).

Теперь обсудим каждое из этих свойств более подробно.

8.12.1. Метки

Для меток, определяющих различные типы частиц, я буду использовать латинские буквы $\{a, b, c, \ldots\}$. (В случае неабелева сверхпроводника метка $(\alpha, R^{(\alpha)})$ обозначала класс сопряженных элементов и неприводимое представление нормализатора класса, но теперь наше обозначение будет более компактным.) Предположим, что совокупность возможных меток конечна. Символ a представляет величину сохраняющегося заряда, переносимого частицей. Иногда мы говорим, что эта метка определяет *суперотборный сектор* теории. Этот термин означает лишь то, что метка a является свойством локализованного объекта, которое нельзя изменить локальным физическим процессом, то есть если одна частица всегда хорошо изолирована от других, ее метка никогда не изменится. В частности, никакие локальные взаимодействия частицы с ее окружением не могут изменить ее метку. Это локальное сохранение заряда является основной причиной того, почему анионы подходят для отказоустойчивой обработки квантовой информации.

Существует одна частная, единичная метка 1. Наличие частицы с меткой 1 фактически эквивалентно отсутствию частицы вообще. Более того, для каждой метки частицы a существует сопряженная метка $\bar{a}$, а также существует операция зарядового сопряжения C (где $C^2 = I$), действие которой отображает метки на сопряженные им:

$$C : a \longmapsto \bar{a} \longmapsto a. \tag{8.64}$$

Метка может быть самосопряженной, так что $\bar{a} = a$. Например, $\bar{1} = 1$.

Нам потребуется рассматривать определенным образом упорядоченные состояния n-частиц. Их удобно представлять расположенными друг за другом вдоль некоторой линии (например, вдоль вещественной оси) слева направо; n частиц помечены как $(a_1, a_2, a_3, \ldots, a_n)$, где a_1 присвоена крайней слева частице, а a_n — крайней справа.

8.12.2. Пространства композитных состояний

Композитный объект, образующийся при объединении двух частиц, также имеет заряд. Возможные значения полного заряда c при значениях зарядов составляющих a и b определяют *правила композиции* модели. Их можно записать как

$$a \times b = \sum_c N_{ab}^c \, c, \tag{8.65}$$

где каждое N_{ab}^c — неотрицательное целое число, а суммирование ведется по полному набору меток c. Отметим, что a, b и c — это метки, а не векторные пространства; произведение слева не является тензорным произведением, а сумма справа не является прямой суммой. Скорее правила композиции можно рассматривать как абстрактное соотношение для набора меток, отображающее упорядоченную тройку $(a, b; c)$ на N_{ab}^c. Это соотношение симметрично по a и b ($a \times b = b \times a$) — возможные заряды композита не зависят от положения a (справа или слева). Прочитанные в обратном направлении, правила композиции определяют возможные способы расщепления заряда c на две части с зарядами a и b.

Если $N_{ab}^c = 0$, то при комбинировании a и b заряд c не может быть получен. Если $N_{ab}^c = 1$, то c *можно* получить единственным способом. Если $N_{ab}^c > 1$, то c можно получить N_{ab}^c различимыми способами. Представление о том, что объединение двух зарядов может дать третий заряд несколькими способами, должно быть знакомо из теории представлений групп. Например, правило, управляющее композицией двух октетных представлений группы SU(3), выглядит следующим образом:

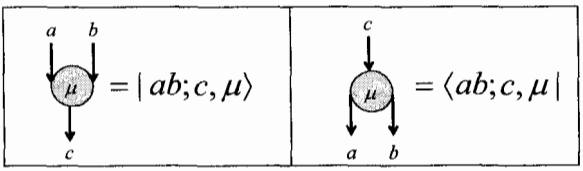
$$8 \times 8 = 1 + 8 + 8 + 10 + \overline{10} + 27, \tag{8.66}$$

так что $N_{88}^{8} = 2$. Но подчеркнем еще раз: в то время как правила композиции представлений групп можно интерпретировать как разложение тензорного произведения векторных пространств на прямую сумму векторных пространств, в общем случае правила композиции в анионной модели такой интерпретации не имеют.

N_{ab}^{c} различимых способов, которыми при объединении a и b может возникнуть c, можно рассматривать как ортонормированные базисные состояния гильбертова пространства V_{ab}^{c}. Мы называем V_{ab}^{c} *пространством композитных состояний*, а состояния, которые оно включает, — *композитными состояниями*. Базисные векторы V_{ab}^{c} можно обозначить как

$$\{|ab; c, \mu\rangle, \quad \mu = 1, 2, \ldots, N_{ab}^{c}\}. \tag{8.67}$$

Для состояний композитного базиса удобно ввести графическое обозначение:



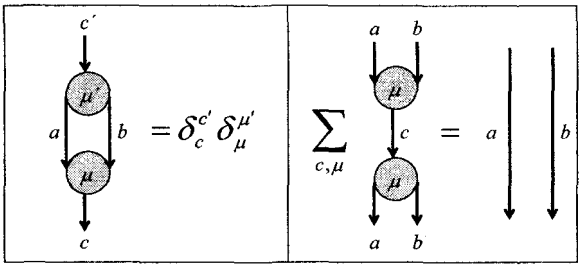
Состояние $|ab; c, \mu\rangle$ представлено в виде окружности, содержащей символ μ; к окружности присоединены обозначенные метками a и b ориентированные входящие линии, представляющие объединяемые заряды, и маркированная меткой c выходящая ориентированная линия, представляющая результат объединения. Существует дуальное векторное пространство V_{c}^{ab}, описывающее состояния, возникающие при расщеплении заряда c на заряды a и b, и дуальный базис с инвертированными линиями (входящей c и выходящими a и b). Пространства V_{ab}^{c} с различными значениями c взаимно ортогональны, так что элементы базиса композитных состояний удовлетворяют уравнению

$$\langle ab; c', \mu' | ab; c, \mu\rangle = \delta_{c}^{c'} \delta_{\mu}^{\mu'}, \tag{8.68}$$

а полноту базиса композитных состояний можно выразить как

$$\sum_{c,\mu} |ab; c, \mu\rangle\langle ab; c, \mu| = I_{ab}, \tag{8.69}$$

где I_{ab} обозначает проектор на пространство $\bigoplus_{c} V_{ab}^{c}$, полное гильбертово пространство для анионной пары ab.



Среди пространств композитных состояний существует несколько естественных изоморфизмов. Во-первых, $V_{ab}^c \cong V_{ba}^c$; эти векторные пространства ассоциируются с различными маркировками двух частиц (при $a \neq b$) и, следовательно, должны рассматриваться как различимые, но они являются изоморфными пространствами, поскольку объединение симметрично. Мы можем также «поднимать и опускать индексы» пространства композитных состояний, заменяя метку ее сопряженной величиной, например,

$$V_{ab}^c \cong V_{a\bar{c}}^{\bar{b}} \cong V_{ab\bar{c}}^1 \cong V_a^{\bar{b}c} \cong V_{\bar{c}}^{\bar{a}\bar{b}} \cong \cdots ; \tag{8.70}$$

на графическом языке этому соответствует обращение соответствующих ориентированных линий с одновременным сопряжением их меток. Пространство $V_{ab\bar{c}}^1$, изображаемое диаграммой с тремя входящими линиями, представляет собой пространство, натянутое на базис различимых способов получения тривиального полного заряда 1 при композиции трех частиц с метками a, b и $\bar{c}$.

Заряд 1 достоин своего имени, поскольку с другими частицами он комбинируется тривиальным образом:

$$a \times 1 = a. \tag{8.71}$$

Вследствие изоморфизма $V_{a1}^a \cong V_{a\bar{a}}^1$, мы делаем вывод, что $\bar{a}$ — единственная метка, которая дает 1 при композиции с a, и что эта композиция может возникнуть лишь единственном способом. Аналогично, $V_{a1}^a \cong V_1^{a\bar{a}}$ означает, что рожденные из вакуума пары частиц имеют сопряженные заряды.

Анионная модель является *неабелевой*, если

$$\dim\left(\bigoplus_c V_{ab}^c\right) = \sum_c N_{ab}^c \geqslant 2 \tag{8.72}$$

по крайней мере для некоторых пар с метками ab; в противном случае модель является *абелевой*. В абелевой модели любые две частицы объеди-

няются единственным способом, а в неабелевой модели существуют некоторые пары частиц, способные объединяться более чем одним способом, и существует гильбертово пространство размерностью два и более, натянутое на эти различимые состояния. Мы будем говорить об этом пространстве как о «топологическом гильбертовом пространстве» анионной пары, подчеркивая тем самым, что данная квантовая информация закодирована нелокально — это коллективное свойство пары, не локализованное ни на какой из образующих ее частиц. Действительно, когда две частицы с метками a и b находятся далеко друг от друга, разные состояния в топологическом гильбертовом пространстве локально выглядят идентичными. Следовательно, эта квантовая информация хорошо спрятана и неуязвима для декогерентизации, обусловленной локальными взаимодействиями с окружающей средой.

Именно по этой причине мы предлагаем использовать неабелевы анионы в работе квантового компьютера. Конечно, нелокально закодированная информация спрятана не только от окружения; она недоступна и для нас. Однако с помощью неабелевых анионов можно убить двух зайцев сразу! По завершении квантового вычисления, когда мы готовы считать результат, мы можем объединить анионы в пары и пронаблюдать результат такой композиции. В сущности, будет достаточно отличать случай, когда заряд композита $c = 1$, от случая $c \neq 1$, то есть отличать остаточную частицу (неспособную распадаться вследствие ее нетривиального сохраняющегося заряда) от отсутствия частицы вообще.

Отметим, что для каждой анионной пары это топологическое гильбертово пространство конечномерно. Обладающая этим свойством анионная модель *рациональна*. Как и в обсуждавшемся случае топологически вырожденного основного состояния абелевой модели, анионы рациональных неабелевых моделей всегда имеют топологические спины, представляющие собой корни из единицы.

8.12.3. Сплетение: R-матрица

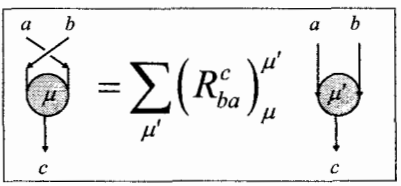
При перестановке против часовой стрелки двух частиц с метками a и b их полный заряд c не изменяется. Следовательно, поскольку две частицы меняются местами на линии, обмен индексами является естественным изоморфизмом, отображающим гильбертово пространство V_{ba}^c на V_{ab}^c; это отображение представляет собой оператор сплетения

$$\mathbf{R} : V_{ba}^c \to V_{ab}^c. \tag{8.73}$$

Если в этих двух пространствах выбрать канонические базисы $\{|ba;c,\mu\rangle\}$ и $\{|ab;c,\mu'\rangle\}$, то операцию (8.73) можно представить как унитарное преобразование

$$\mathbf{R}:|ba;c,\mu\rangle \longmapsto \sum_{\mu'}|ab;c,\mu'\rangle(R^c_{ab})^{\mu'}_{\mu}; \tag{8.74}$$

отметим, что $\mathbf{R}$ может оказывать нетривиальное действие на композитные состояния. Представляя действие $\mathbf{R}$ в графической форме, удобно зафиксировать положения меток a и b на входящих линиях и повернуть против часовой стрелки линии, входящие в вершину композита (μ). График с перекрещенными линиями представляет состояние в пространстве V^c_{ab}, полученное в результате применения матрицы $\mathbf{R}$ к $|ba;c,\mu\rangle$, которое может быть разложено в каноническом базисе пространства V^c_{ab}:



Оператор *монодромии*

$$\mathbf{R}^2 : V^c_{ab} \to V^c_{ab} \tag{8.75}$$

является изоморфизмом V^c_{ab} на самого себя, представляя результат полного оборота a вокруг b против часовой стрелки. Как уже отмечалось при обсуждении неабелева сверхпроводника, оператор монодромии эквивалентен повороту c на 2π с одновременным поворотом a и b на -2π; следовательно, собственные значения оператора монодромии определяются *топологическими спинами* частиц:

$$(R^c_{ab})^2 = e^{-2\pi i J_c} e^{2\pi i J_a} e^{2\pi i J_b} \equiv e^{i(\theta_c - \theta_a - \theta_b)}. \tag{8.76}$$

Более того, как и в случае абелевых анионов, топологический спин определяется оператором сплетения, действующим на пару частица-античастица с тривиальным полным зарядом:

$$e^{-i\theta_a} = R^1_{a\bar{a}} \tag{8.77}$$

(поскольку рождение пары, перестановка и аннигиляция пары эквивалентны повороту частицы на -2π).

8.12.4. Ассоциативность композитных состояний: F-матрица

Композитные состояния ассоциативны:

$$(a \times b) \times c = a \times (b \times c). \tag{8.78}$$

С математической точки зрения, это аксиома, которой удовлетворяют правила композиции анионной модели. С физической точки зрения, это необходимое условие, поскольку полный заряд системы из трех частиц — это ее внутреннее свойство, которое не должно зависеть от того, объединяются ли сначала a и b, а затем эта пара — с c, или же сначала комбинируются b и c, а затем результат композиции — с a.

Следовательно, топологическое гильбертово пространство композитных состояний с полным зарядом d, образующихся при объединении трех частиц с зарядами a, b и c, допускает два естественных способа разложения в прямую сумму тензорных произведений пространств композитных состояний пар частиц:

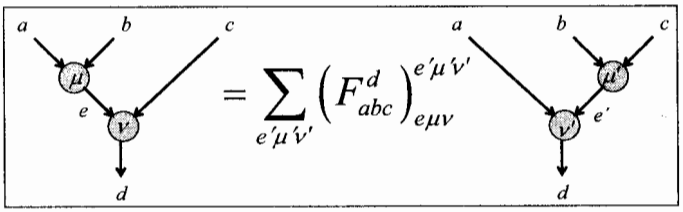
$$V_{abc}^{d} \cong \bigoplus_{e} V_{ab}^{e} \otimes V_{ec}^{d} \cong \bigoplus_{e'} V_{ae'}^{d} \otimes V_{bc}^{e'}. \tag{8.79}$$

Соответственно, в пространстве V_{abc}^{d} существует два естественных ортонормированных базиса

$$\begin{aligned} |(ab)c \to d; e\mu\nu\rangle &\equiv |ab; e, \mu\rangle \otimes |ec; d, \nu\rangle, \\ |a(bc) \to d; e'\mu'\nu'\rangle &\equiv |ae'; d, \nu'\rangle \otimes |bc; e', \mu'\rangle, \end{aligned} \tag{8.80}$$

связанных между собой унитарным преобразованием $\mathbf{F}$:

$$|(ab)c \to d; e\mu\nu\rangle = \sum_{e'\mu'\nu'} |a(bc) \to d; e'\mu'\nu'\rangle \left(F_{abc}^{d}\right)_{e\mu\nu}^{e'\mu'\nu'}. \tag{8.81}$$



Унитарные матрицы F_{abc}^{d} иногда называют *матрицами композиции*; однако чтобы не рисковать вызвать путаницу между ними и правилами композиции N_{ab}^{c}, я лучше буду называть их *F-матрицами*.

8.12.5. Множество анионов: стандартный базис

В анионном квантовом компьютере мы оперируем с n-анионным топологическим квантовым состоянием путем сплетения анионов. Чтобы описывать эти вычисления, удобно выбрать стандартный базис в таком гильбертовом пространстве.

Пусть n анионов с полным зарядом c, расположенные последовательно вдоль линии, имеют метки $a_1, a_2, a_3, \ldots, a_n$. Рассмотрим процесс,в котором сначала происходит объединение анионов 1 и 2, затем образовавшаяся пара объединяется с анионом 3, тройка — с анионом 4, и так далее. Совершаемому в этом порядке объединению соответствует разложение n-анионного топологического гильбертова пространства:

$$V^c_{a_1a_2a_3\cdots a_n} \cong \bigoplus_{b_1,b_2,\ldots,b_{n-2}} V^{b_1}_{a_1a_2} \otimes V^{b_2}_{b_1a_3} \otimes V^{b_3}_{b_2a_4} \otimes \cdots \otimes V^c_{b_{n-2}a_n}. \quad (8.82)$$

Отметим, что это пространство *не имеет* естественного разложения на тензорное произведение подсистем, соответствующих локализованным частицам; скорее, мы представили его как прямую сумму множества тензорных произведений. Для неабелевых анионов его размерность экспоненциальна по n:

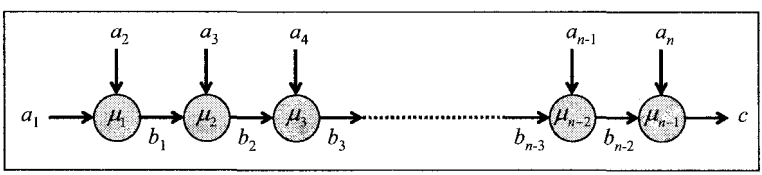
$$\dim(V^c_{a_1a_2a_3\cdots a_n}) \equiv N^c_{a_1a_2a_3\cdots a_n} = \\ = \sum_{b_1,b_2,b_3,\ldots b_{n-2}} N^{b_1}_{a_1a_2} N^{b_2}_{b_1a_3} N^{b_3}_{b_2a_4} \cdots N^c_{b_{n-2}a_n}. \quad (8.83)$$

Таким образом, топологическое гильбертово пространство является подходящей ареной для успешной обработки квантовой информации.

С этим разложением ассоциируется стандартный базис, элементы которого нумеруются промежуточными зарядами $b_1, b_2, \ldots, b_{n-2}$ и представляют собой произведения базисных векторов $\{|\mu_j\rangle\}$ пространств композитных состояний $V^{b_j}_{b_{j-1},a_{j+1}}$:

$$\{|a_1a_2; b_1, \mu_1\rangle|b_1a_3; b_2, \mu_2\rangle \cdots |b_{n-3}a_{n-1}; b_{n-2}, \mu_{n-2}\rangle|b_{n-2}a_n; c, \mu_{n-1}\rangle\}, \quad (8.84)$$

или на графическом языке:



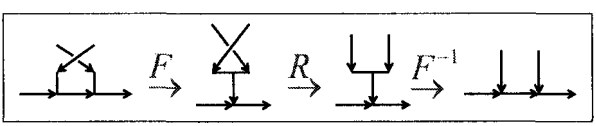
Конечно, выбор этого базиса неоднозначен. При желании можно представить процесс объединения частиц, протекающий в другом порядке, и получить другой стандартный базис, связанный с предыдущим унитарной F-матрицей.

8.12.6. Сплетение в стандартном базисе: B-матрица

Рассмотрим, что произойдет с состояниями топологического векторного пространства n анионов $V^c_{a_1a_2a_3\cdots a_n}$ при перестановке частиц друг с другом. Поскольку перестановки могут менять позиции частиц с различными метками, они могут отображать одно топологическое векторное пространство на другое путем перестановки меток. Тем не менее, можно рассмотреть прямые суммы векторных пространств, ассоциируемых со всеми возможными перестановками меток, представляющими группу кос B_n.

Попробуем описать, как это представление действует на стандартные базисы в этих пространствах. Достаточно сказать, как представляются перестановки соседних частиц, то есть определить действие генераторов группы кос. Однако до настоящего времени мы обсуждали действие группы кос лишь на пару частиц с определенным полным зарядом (R-матрица), что по сути своей не дает информации о ее действии на стандартные базисы.

Выход из этого затруднительного положения дает следующее наблюдение: применяя F-матрицу, можно перейти от стандартного базиса к базису, в котором R-матрица блочно-диагональна, применить $\mathbf{R}$, а затем, применяя $\mathbf{F}^{-1}$, вернуться в стандартный базис:



Композиция этих трех операций, выражающая результат сплетения в стандартном базисе, обозначается символом $\mathbf{B}$ и иногда называется «матрицей сплетения»; но чтобы избежать путаницы между $\mathbf{B}$ и $\mathbf{R}$, я буду называть ее просто *B-матрицей*.

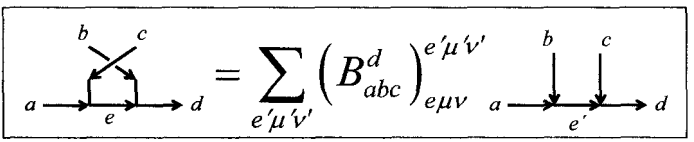
Рассмотрим перестановку анионов в позициях j и $j+1$ вдоль линии. В нашем разложении пространства $V^c_{a_1a_2a_3\cdots a_n}$ эта перестановка действует на пространство

$$V^{b_j}_{b_{j-2},a_j,a_{j+1}} = \bigoplus_{b_{j-1}} V^{b_{j-1}}_{b_{j-2},a_j} \otimes V^{b_j}_{b_{j-1},a_{j+1}}. \tag{8.85}$$

Сокращая число подстрочных индексов, будем называть это пространство V^d_{acb}, которое преобразуется перестановкой как

$$\mathbf{B} : V^d_{acb} \to V^d_{abc}. \tag{8.86}$$

Выразим действие $\mathbf{B}$ в стандартных базисах двух пространств V^d_{acb} и V^d_{abc}:



Чтобы избежать нагромождения в уравнениях, я опускаю метки для базисных элементов пространства композитных состояний (расположение этих меток очевидно). Следовательно, мы пишем

$$\begin{aligned}
\mathbf{B}|(ac)b \to d; e\rangle &= \sum_f \mathbf{B}|a(cb) \to d; f\rangle \left(F^d_{acb}\right)^f_e = \\
&= \sum_f |a(bc) \to d; f\rangle R^f_{bc} \left(F^d_{acb}\right)^f_e = \\
&= \sum_{f,g} |(ab)c \to d; g\rangle \left[(F^{-1})^d_{abc}\right]^g_f R^f_{bc} \left(F^d_{acb}\right)^f_e ,
\end{aligned} \tag{8.87}$$

или

$$\mathbf{B} : |(ac)b \to d; e\rangle \longmapsto \sum_g |(ab)c \to d; g\rangle \left(B^d_{abc}\right)^g_e , \tag{8.88}$$

где

$$\left(B^d_{abc}\right)^g_e = \sum_f \left[(F^{-1})^d_{abc}\right]^g_f R^f_{bc} \left(F^d_{acb}\right)^f_e . \tag{8.89}$$

Мы, как и хотели, выразили действие B-матрицы в стандартном базисе через F- и R-матрицы.

Таким образом, реализованное n анионами представление группы кос полностью характеризуется F-матрицей и R-матрицей. Более того, мы увидели, что R-матрица также определяет топологические спины анионов, так что мы фактически построили представление более широкой группы, генераторы которой включают как перестановки соседних частиц, *так и* повороты частиц на 2π. Подходящим названием для этой группы было бы «группа лент», поскольку ее элементы скорее находятся во взаимно однозначном

соответствии с топологическими классами сплетенных лент (которые могут скручиваться), нежели сплетенных струн; однако математики уже назвали это «группой классов отображений для сферы с n проколами».

На данном этапе мы заканчиваем наше общее описание анионной модели. Модель определяется следующими параметрами: (1) набор меток, (2) правила композиции, (3) R-матрица и (4) F-матрица.

Построенный нами математический объект называется *унитарным топологическим модулярным функтором*. Он тесно связан с двумя другими объектами, которые уже были хорошо изучены: *теорией топологического квантового поля* в $2+1$-мерном пространстве-времени, а также *теорией конформного поля* в $1+1$-мерном пространстве-времени. Однако мы будем называть его просто *анионной моделью*.

8.13. Моделирование анионов квантовой схемой

Топологические квантовые вычисления выполняются в три этапа:

1. *Инициализация*: Создаются пары частица-античастица $c_1\bar{c}_1$, $c_2\bar{c}_2$, $c_3\bar{c}_3$, $\ldots, c_m\bar{c}_m$. Каждая пара определенного типа и с тривиальным полным зарядом.

2. *Обработка*: Количество $n = 2m$ частиц проводятся вдоль траекторий, их мировые линии следуют определенной косе.

3. *Считывание*: Пары оказавшихся рядом частиц объединяются и ведется запись, аннигилирует каждая такая пара полностью или нет. Эта запись является результатом вычислений.

(В случае вычислительной модели неабелева сверхпроводника предполагалось, что сплетение определяется результатом объединения, выполняемого на стадии обработки. Но здесь рассматривается модель, в которой все измерения откладываются вплоть до финального считывания).

Какой мощностью обладает данная модель вычислений? Я утверждаю, что этот топологический квантовый компьютер можно эффективно смоделировать с помощью квантовой схемы, поскольку топологическое гильбертово пространство n-анионов не имеет простого и естественного разложения на тензорное произведение малых подсистем. Возможно, это утверждение непосредственно не очевидно. Для доказательства, необходимо объяснить следующее:

1. как с помощью обычных кубитов закодировать топологическое гильбертово пространство,

2. как, используя квантовые вентили, эффективно представить сплетение,

3. как смоделировать объединение анионной пары.

Кодирование. Поскольку каждая образованная в процессе инициализации пара имеет тривиальный полный заряд, исходное n-анионное состояние также имеет тривиальный полный заряд. Следовательно, для определенного набора меток $a_1, a_2, a_3, \ldots, a_n$ топологическое гильбертово пространство имеет вид

$$V^1_{a_1 a_2 a_3 \cdots a_n} \cong \bigoplus_{b_1, b_2, \ldots, b_{n-3}} V^{b_1}_{a_1 a_2} \otimes V^{b_2}_{b_1 a_3} \otimes \cdots \otimes V^{\bar{a}_n}_{b_{n-3} a_{n-1}}; \tag{8.90}$$

имеется $n-3$ промежуточных зарядов и $n-2$ пространств композитных состояний, возникающих в каждом слагаемом. Перестановки частиц меняют местами метки, но после каждой перестановки векторное пространство попрежнему имеет вид (8.90), отличаясь от него лишь порядком следования меток a_j.

Хотя каждое n-анионное топологическое гильбертово пространство само по себе не является тензорным произведением подсистем, все эти пространства содержатся в

$$(\mathcal{H}_d)^{\otimes(n-2)}, \tag{8.91}$$

где

$$\mathcal{H}_d = \bigoplus_{a,b,c} V^1_{abc}. \tag{8.92}$$

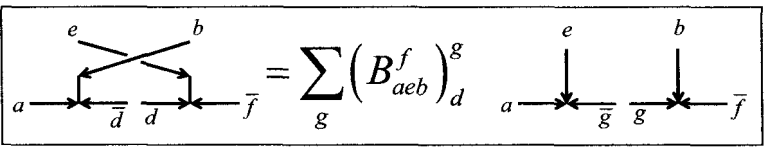
Здесь a, b, c суммируются по полному набору меток модели (которая по предположению является конечной), так что пространство $\mathcal{H}_d$ содержит все возможные композитные состояния трех частиц, а его размерность d равна

$$d = \sum_{a,b,c} N^1_{abc}. \tag{8.93}$$

Таким образом, n-анионное состояние можно закодировать в гильбертовом пространстве $(n-2)$ *кудитов* некоторой постоянной размерности d (которая зависит от анионной модели, но не зависит от n). Базисные состояния этого кудита можно задать в виде $\{|a, b, c; \mu\rangle\}$, где μ маркирует элемент базиса пространства композитных состояний V^1_{abc}.

Сплетение. В топологическом квантовом компьютере коса сплетается путем выполнения ряда перестановок, каждая из которых действует на пару соседних частиц. Результат каждой перестановки в стандартном базисе

ывается B-матрицей. Как она действует в закодированном (с помощью тов) топологическом векторном пространстве? Опуская для краткости и композитных состояний, наш базис для двухкудитовых состояний но обозначить как $|a,b,c\rangle|d,e,\bar{f}\rangle$. Но в топологическом квантовом компьютере метки d и $\bar{c}$ всегда совпадают, и, следовательно, для выполнения елирования сплетения нужно рассмотреть лишь такие двухкудитовые ояния, метки которых совпадают в следующем смысле:

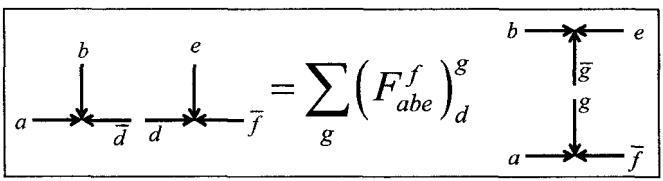


а действие B-матрицы на эти базисные состояния имеет вид

$$\mathbf{B} : |a,b,\bar{d}\rangle|d,e,\bar{f}\rangle \longmapsto \sum_g |a,e,\bar{g}\rangle|g,b,\bar{f}\rangle \left(B^f_{aeb}\right)^g_d . \qquad (8.94)$$

и требовалось, мы представили $\mathbf{B}$ как $d^2 \times d^2$-матрицу, действующую ару соседних кудитов.

Объединение. Объединение анионной пары можно смоделировать кудитовым измерением, которое с помощью F-матрицы можно свести нокудитовому измерению:



мотрим базисное состояние $|a,b,\bar{d}\rangle|d,e,\bar{f}\rangle$ пары соседних кудитов; ка- амплитуда вероятности того, что анионная пара (be) имеет тривиный полный заряд? Используя F-преобразование, ее состояние можно ставить в виде разложения

$$\begin{aligned} \mathbf{F} : |a,b,\bar{d}\rangle|d,e,\bar{f}\rangle &\longmapsto \sum_g |a,g,\bar{f}\rangle|b,\bar{g},e\rangle \left(F^f_{abe}\right)^g_d = \\ &= |a,1,\bar{f}\rangle|b,1,e\rangle \left(F^f_{abe}\right)^1_d + \sum_{g\neq 1} |a,g,\bar{f}\rangle|b,\bar{g},e\rangle \left(F^f_{abe}\right)^g_d ; \end{aligned} \qquad (8.95)$$

мы явно выделили из суммы по g слагаемое, отвечающее композитному состоянию с тривиальным зарядом 1. После F-преобразования (которое является лишь специфическим двухкудитовым унитарным вентилем) можно определить вероятность того, что (be) комбинируется к 1, выполняя проекционное измерение второго кудита в базисе $\{|b, \bar{g}, e\rangle\}$, и записывая, действительно ли $g = 1$.

Этим завершается доказательство того, что квантовая схема может эффективно моделировать топологический квантовый компьютер.

8.14. Анионы Фибоначчи

Мы установили, что топологические квантовые вычисления не мощнее модели квантовой схемы — любая задача, которую можно эффективно решить путем сплетения неабелевых анионов, может быть также эффективно решена с помощью квантовой схемы. Но насколько мощными являются топологические вычисления? Можно ли с помощью сплетения анионов смоделировать универсальный квантовый компьютер? Ответ зависит от специфических свойств анионов: одни модели неабелевых анионов универсальны, другие — нет. Чтобы найти ответ для конкретной анионной модели, необходимо понять свойства представлений группы кос, которые определяются F- и R-матрицами.

Вместо того чтобы заниматься общим обсуждением, изучим одну особенно простую модель неабелевых анионов и продемонстрируем ее вычислительную универсальность. Это простейшая из неабелевых моделей — специалисты в области конформной теории поля называют ее «моделью Янга–Ли», но я буду называть ее «моделью Фибоначчи» по причинам, которые вскоре станут ясны.

В модели Фибоначчи имеется лишь две метки: тривиальная метка, которую я сейчас обозначу как 0, и единственная нетривиальная метка, которую я обозначу как 1, где $\bar{1} = 1$. И существует только одно нетривиальное правило композиции:

$$1 \times 1 = 0 + 1; \tag{8.96}$$

при столкновении два аниона либо аннигилируют, либо превращаются в изолированный анион. Модель неабелева, поскольку два аниона могут объединяться двумя различимыми способами.

Рассмотрим стандартный базис для n-анионного гильбертова пространства V_1^{bn}, каждый базисный элемент которого описывает различимый способ, которым могут комбинироваться n анионов, выдавая в итоге полный заряд $b \in \{0, 1\}$. Пусть сначала объединяются два крайних слева ани-

она; зарядом возникающего в результате состояния может быть 0 или 1. Затем этот заряд объединяется с третьим слева анионом, давая полный заряд 0 или 1, и так далее. Наконец, последний анион объединяется с полным зарядом первых $n-1$ анионов и выдает полный заряд b. В этом описании процесса появляется всего $n-2$ промежуточных зарядов $b_1, b_2, b_3, \ldots, b_{n-2}$; таким образом, соответствующие элементы базиса можно обозначить двоичной строкой длины $n-2$. Если полный заряд равен 0, результат композиции первых $n-1$ анионов *должен* быть 1, так что базисные состояния маркируются строками длиной $n-3$.

Однако *не все* бинарные строки допустимы — за 0 всегда должна следовать 1. Не существует строк, содержащих два нуля подряд, поскольку единственно возможным результатом объединения зарядов 0 и 1 является полный заряд 1. Во всем остальном, в последовательности не имеется ограничений. Следовательно, базисные состояния находятся во взаимно однозначном соответствии с двоичными строками, не содержащими ни одной пары следующих друг за другом нулей.

Таким образом, размерность $N_n^0 \equiv N_{1^n}^0$ топологического гильбертова пространства $V_{1^n}^0$ подчиняется простому рекуррентному соотношению. Если объединение первых двух частиц дает тривиальный полный заряд, то оставшиеся $n-2$ частиц могут объединяться N_{n-2}^0 различимыми способами. Если же объединение первых двух частиц дает анион с нетривиальным зарядом, то этот анион может объединиться с другими $n-2$ анионами N_{n-1}^0 способами; следовательно,

$$N_n^0 = N_{n-1}^0 + N_{n-2}^0. \tag{8.97}$$

Поскольку $N_1^0 = 0$ и $N_2^0 = 1$, решением данного рекуррентного соотношения является

$$\begin{array}{rcccccccccc} n & = & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9\ldots \\ N_n^0 & = & 0 & 1 & 1 & 2 & 3 & 5 & 8 & 13 & 21\ldots \end{array} \tag{8.98}$$

— последовательность чисел Фибоначчи (вот почему я называю эту модель «моделью Фибоначчи»).

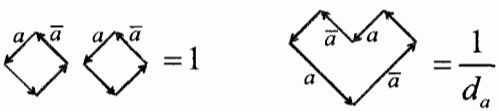
Числа Фибоначчи растут вместе с n со скоростью $N_n^0 \approx C\phi^n$, где ϕ — золотое сечение $\phi = (1+\sqrt{5})/2 \approx 1.618$. Поскольку ϕ управляет скоростью, с которой расширяется гильбертово пространство при добавлении анионов, мы говорим, что $d = \phi$ — *квантовая размерность* аниона Фибоначчи. Иррациональность этой «размерности» служит яркой иллюстрацией того, что топологическое гильбертово пространство не имеет естественного разложения на тензорное произведение подсистем — напротив, топологически

закодированная квантовая информация является коллективным свойством n анионов.

8.15. Квантовая размерность

Мы еще вернемся к свойствам модели Фибоначчи, но сначала более глубоко изучим концепцию квантовой размерности. Как определить размерность d_a метки a для общей анионной модели? С этой целью удобно представить физический процесс, в котором рождаются две пары $a\bar{a}$ (каждая с тривиальным полным зарядом); затем частица a из пары справа объединяется с античастицей $\bar{a}$ из пары слева. Аннигилируют ли эти частицы?

При соответствующем соглашении относительно фаз, *амплитуда* вероятности аннигиляции является действительным числом из единичного интервала [0,1]. Определим его как $1/d_a$, где d_a — квантовая размерность a (а $1/d_a^2$ — *вероятность* аннигиляции). Отметим, что из этого определения становится очевидным равенство $d_a = d_{\bar{a}}$. В том случае, когда метка a является меткой неприводимого представления R_a группы G, ее размерность в точности равна $d_a = |R_a|$, то есть размерности представления. Графически это понять проще:



Если рождаются две пары, а затем каждая пара немедленно аннигилирует, их мировые линии образуют две замкнутые петли, а $|R|$ подсчитывает количество распространяющихся вдоль каждой петли различимых «цветов». Но если частица из каждой пары аннигилирует с античастицей другой пары, то существует только одна замкнутая петля и, следовательно, одно суммирование по цветам; если мы нормируем процесс слева на единицу, амплитуда для процесса справа подавляется коэффициентом $1/|R|$. Выражая эту же идею на языке уравнений, запишем нормированное состояние $R\bar{R}$-пары в виде

$$|R\bar{R}\rangle = \frac{1}{\sqrt{|R|}} \sum_i |i\rangle|\bar{i}\rangle, \tag{8.99}$$

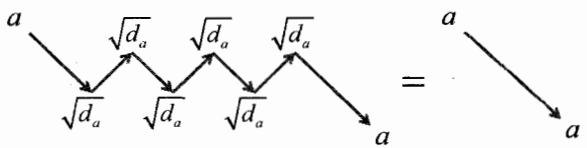
где $\{|i\rangle\}$ обозначает ортонормированный базис для R, а $\{|\bar{i}\rangle\}$ — базис для $\bar{R}$. Предположим, что рождаются пары $|R\bar{R}\rangle$ и $|R'\bar{R}'\rangle$; если после обмена эле-

ментами пары объединяются, то амплитуда аннигиляции равна

$$\langle R\bar{R}, R'\bar{R}'|R\bar{R}', R'\bar{R}\rangle = \frac{1}{|R|^2}\sum_{i,i',j,j'}\langle j\bar{j}, j'\bar{j}'|i\bar{i}', i'\bar{i}\rangle = \\ = \frac{1}{|R|^2}\sum_{i,i',j,j'}\delta_{ji}\delta_{ji'}\delta_{j'i'}\delta_{j'i} = \frac{1}{|R|^2}\sum_{i}\delta_{ii} = \frac{1}{|R|}. \tag{8.100}$$

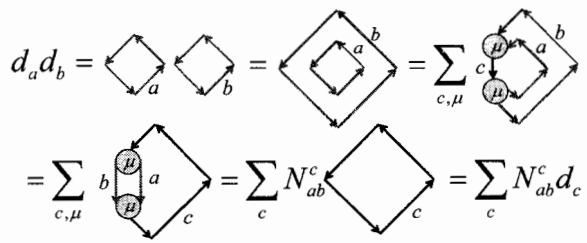
Однако в общем случае квантовая размерность не имеет прямой интерпретации как «счетчика цветов» и нет причин, по которым она должна быть целым числом.

Как такие квантовые размерности связаны с размерностями топологических гильбертовых пространств? Чтобы увидеть эту связь, полезно изменить наши соглашения о нормировке. Обратите внимание на то, что, создавая множество пар $a\bar{a}$ и аннигилируя частицу каждой пары с античастицей следующей, в мировую линию аниона a можно ввести множество «зигзагов». Но каждый зигзаг понижает амплитуду на очередной множитель $1/d_a$. Его можно скомпенсировать, приписывая каждому рождению и аннигиляции пары весовой весовой множитель $\sqrt{d_a}$. С учетом этого соглашения мы можем, не опасаясь неприятных последствий, изгибать мировую линию частицы вперед и назад во времени:



Теперь приписанный мировой линии вес является топологическим инвариантом (при деформации линии он остается неизменным), а мировая линия частицы типа a, образующая замкнутую петлю, приобретает весовой множитель d_a.

С учетом этих новых соглашений можно объяснить следующую последовательность действий:



Каждая диаграмма представляет скалярное произведение двух (нормированных нестандартно) состояний. Мы вставили сумму по всем меткам (c) и соответствующим композитным состояниям (μ), которые могут возникнуть при объединении a и b. Затем, используя топологическую инвариантность диаграммы, мы вывернули ее наизнанку и сократили композитные состояния (приобретая множитель N_{ab}^{c}, подсчитывающий возможные значения μ).

Выведенное нами уравнение

$$d_a d_b = \sum_c N_{ab}^{c} d_c \equiv \sum_c (N_a)_b^c d_c \tag{8.101}$$

означает, что вектор $\vec{d}$, компонентами которого являются квантовые размерности, является собственным вектором с собственным значением d_a матрицы N_a, описывающей правила композиции метки a с другими метками:

$$N_a \vec{d} = d_a \vec{d}. \tag{8.102}$$

Более того, поскольку матрица N_a имеет неотрицательные элементы, а все компоненты вектора $\vec{d}$ положительны, размерность d_a равна наибольшему собственному значению матрицы N_a и является невырожденной. (Это простое замечание иногда называют *теоремой Перрона–Фробениуса*.) Для n анионов с одинаковыми метками a топологическое гильбертово пространство $V_{aaa\cdots a}^{b}$ сектора с полным зарядом b имеет размерность

$$N_{aaa\cdots a}^{b} = \sum_{\{b_i\}} N_{aa}^{b_1} N_{ab_1}^{b_2} N_{ab_2}^{b_3} \cdots N_{ab_{n-2}}^{b} = \langle b|(N_a)^{n-1}|a\rangle. \tag{8.103}$$

Матрица N_a может быть диагонализована и представлена в виде

$$N_a = |v\rangle d_a \langle v| + \cdots, \tag{8.104}$$

где

$$|v\rangle = \frac{\vec{d}}{\mathcal{D}}, \quad \mathcal{D} = \sqrt{\sum_c {d_c}^2}, \tag{8.105}$$

и опущены собственные значения более высоких порядков малости; следовательно:

$$N_{aaa\cdots a}^{b} = d_a^n d_b / \mathcal{D}^2 + \cdots, \tag{8.106}$$

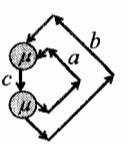
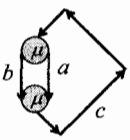
где многоточие обозначает слагаемые, экспоненциально малые при больших n. Таким образом, квантовая размерность d_a определяет скорость роста n-частичного гильбертова пространства для анионов типа a.

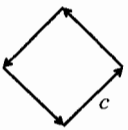
Поскольку метка 0 с тривиальным зарядом комбинируется тривиально, мы имеем $d_0 = 1$. В случае модели Фибоначчи из правила композиции $1 \times 1 = 0 + 1$ следует, что $d_1^2 = 1 + d_1$, решением которого является $d_1 = \phi$, как мы обнаружили ранее; следовательно, $\mathcal{D}^2 = d_0^2 + d_1^2 = 1 + \phi^2 = 2 + \phi$. Наша формула принимает вид

$$N_{111\cdots1}^0 = \left(\frac{1}{2+\phi}\right)\phi^n, \tag{8.107}$$

что является прекрасным приближением к числам Фибоначчи даже при умеренных значениях n.

Предположим, что одновременно рождаются пары $a\bar{a}$ и $b\bar{b}$. Какова вероятность $p(ab \to c)$ того, что при объединении частиц a и b возникнет состояние с полным зарядом c? На этот вопрос можно ответить, используя те же графические манипуляции:

$d_a d_b p(ab \to c) = \sum_\mu$  $= \sum_\mu$ 

$= N_{ab}^c$  $= N_{ab}^c d_c$

Деля на $d_a d_b$ с целью восстановить подходящую нормировку скалярного произведения, мы приходим к выводу, что

$$p(ab \to c) = \frac{N_{ab}^c d_c}{d_a d_b}, \tag{8.108}$$

что обобщает формулу $p(a\bar{a} \to 1) = 1/d_a^2$, которую мы использовали для определения квантовой размерности, и удовлетворяет условию нормировки

$$\sum_c p(ab \to c) = 1. \tag{8.109}$$

Чтобы прийти к другой интерпретации квантовой размерности, представим рождение плотного анионного газа, который затем некоторое время

отжигается — анионы сталкиваются и объединяются, постепенно понижая количество частиц. В конечном счете (но задолго до достижения теплового равновесия) частота столкновений становится настолько низкой, что процессы объединения практически останавливаются. К этому моменту, каким бы ни было начальное распределение типов частиц, достигается устойчивое распределение состояний, которое сохраняется столкновениями. Если в устойчивом состоянии частицы типа a появляются с вероятностью p_a, то

$$\sum_{ab} p_a p_b p(ab \to c) = p_c. \tag{8.110}$$

Используя уравнение

$$\sum_a N_{ab}^c d_a = \sum_a N_{b\bar{c}}^a \, d_{\bar{a}} = d_b d_{\bar{c}} = d_b d_c, \tag{8.111}$$

нетрудно убедиться в том, что это условие удовлетворяется уравнением

$$p_a = \frac{d_a^2}{\mathcal{D}^2}. \tag{8.112}$$

Таким образом, в случайном процессе рождения анионов бóльшую вероятность генерации имеют анионы, несущие метки бóльшей квантовой размерности, в соответствии с тем свойством, что анионы с бóльшей размерностью имеют больше квантовых состояний.

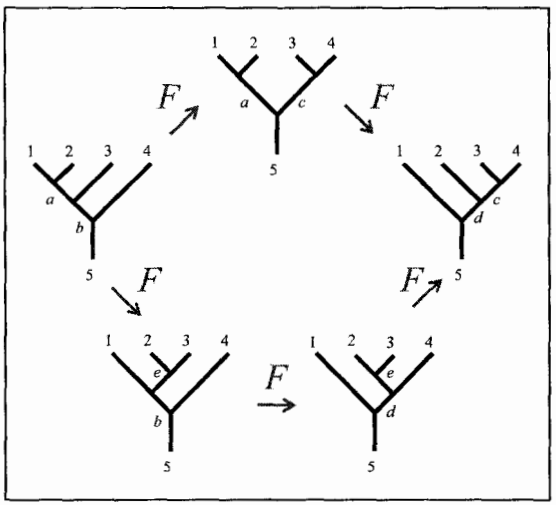
8.16. Уравнения пяти- и шестиугольника

Чтобы оценить вычислительную мощность анионной модели (такой, как модель Фибоначчи), необходимо знать свойства сплетений анионов, которые определяются R- и F-матрицами. Мы увидим, что правила сплетения существенно ограничиваются алгебраическими условиями совместимости. В случае модели Фибоначчи этих условий согласования достаточно для определения единственного правила сплетения, совместимого с правилами композиции.

Условия совместимости возникают потому, что изоморфизм, связывающий два топологических пространства, можно получить, выполняя последовательность «F-преобразований» и «R-преобразований». Изоморфизм можно рассматривать как унитарную матрицу, которая связывает канонические ортонормированные базисы двух разных пространств; это унитар-

ное преобразование не зависит от конкретной последовательности преобразований, из которых конструируется изоморфизм, а лишь от начального и конечного базисов.

Например, существует пять связанных F-преобразованиями разных способов объединения четырех частиц (без их перестановок):



Изображенный крайним слева на этой пятиугольной схеме базис — «левый стандартный базис» $\{|\text{left}; a, b\rangle\}$, в котором сначала объединяются частицы 1 и 2, результирующий заряд a объединяется с частицей 3 и в результате выдает заряд b, а затем, наконец, b объединяется с частицей 4 и дает полный заряд 5. Изображенный крайним справа базис — «правый стандартный базис» $\{|\text{right}; c, d\rangle\}$, в котором частицы объединяются не слева направо, а наоборот — справа налево. Через вершину пятиугольника эти два базиса связаны двумя F-преобразованиями, то есть

$$|\text{left}; a, b\rangle = \sum_{c,d} |\text{right}; c, d\rangle (F^5_{12c})^d_a (F^5_{a34})^c_b. \tag{8.113}$$

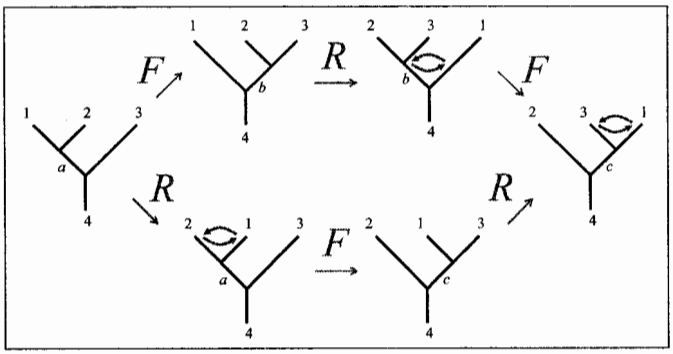
Через нижнюю часть пятиугольника они связаны тремя F-преобразованиями, другими словами,

$$|\text{left}; a, b\rangle = \sum_{c,d,e} |\text{right}; c, d\rangle (F^d_{234})^c_e (F^5_{1e4})^d_b (F^b_{123})^e_a. \tag{8.114}$$

Сравнение этих двух выражений для $|\text{left}; a, b\rangle$ дает *уравнение пятиугольника*:

$$(F^5_{12c})^d_a (F^5_{a34})^c_b = \sum_e (F^d_{234})^c_e (F^5_{1e4})^d_b (F^b_{123})^e_a. \tag{8.115}$$

Другое нетривиальное условие совместимости можно получить, если рассмотреть различные способы объединения трех частиц:



Крайний слева на этой шестиугольной схеме базис $\{|\text{left}; a\rangle\}$ получается, если частицы располагаются в порядке 123 и сначала объединяются частицы 1 и 2. Крайний справа базис $\{|\text{right}; c\rangle\}$ получается при расстановке частиц в порядке 231, и если сначала комбинируются частицы 1 и 3. Через верхнюю часть шестиугольника два этих базиса связаны последовательностью преобразований FRF:

$$|\text{left}, a\rangle = \sum_{b,c} |\text{right}; c\rangle (F^4_{231})^c_b R^4_{1b} (F^4_{123})^b_a. \tag{8.116}$$

Через нижнюю часть шестиугольника они связаны последовательностью преобразований RFR, то есть

$$|\text{left}, a\rangle = \sum_c |\text{right}; c\rangle R^c_{13} (F^4_{213})^c_a R^a_{12}. \tag{8.117}$$

Сравнение этих двух выражений для $|\text{left}, a\rangle$ дает *уравнение шестиугольника*:

$$R^c_{13} (F^4_{213})^c_a R^a_{12} = \sum_b (F^4_{231})^c_b R^4_{1b} (F^4_{123})^b_a. \tag{8.118}$$

Красивая теорема, которую здесь я не стану доказывать, утверждает, что не существует иных условий, обеспечивающих совместимость сплетения и объединения. То есть при любом выборе начального и конечного базисов для n анионов *все* последовательности R- и F-преобразований, переводящих исходный базис в конечный, дают *один и тот же* изоморфизм при условии справедливости уравнений пяти- и шестиугольника. Данная теорема является частным случаем *теоремы когерентности Маклейна*, фундаментального результата теории категорий. Вместе уравнения пяти- и шестиугольника называются *полиномиальными уравнениями Мура–Сейберга* — их важность для физики была впервые осознана в 80-е годы при изучении (1+1)-мерной конформной теории поля.

Решение полиномиальных уравнений определяет жизнеспособные анионные модели. Следовательно, существует систематическая процедура их конструирования:

1. выбрать набор меток и принять правило композиции,
2. решить полиномиальные уравнения для $\mathbf{R}$ и $\mathbf{F}$.

Если решений не существует, то гипотетическое правило композиции несовместимо с принципами локальной квантовой физики и должно быть забраковано. Если существует более одного решения (причем решения не связаны друг с другом перестановкой меток, переопределением базисов и т. п.), то каждое решение определяет отдельную модель с принятым правилом композиции.

Чтобы проиллюстрировать эту процедуру, рассмотрим полиномиальные уравнения для правила композиции Фибоначчи. Здесь фигурируют только две F-матрицы, которые мы будем обозначать как

$$\mathbf{F}_{0111} \equiv \mathbf{F}_0, \quad \mathbf{F}_{1111} \equiv \mathbf{F}_1. \tag{8.119}$$

$\mathbf{F}_0$ в действительности является матрицей размерности 1×1

$$(F_0)_a^b = \delta_a^1 \delta_1^b, \tag{8.120}$$

тогда как $\mathbf{F}_1$ — матрица размерности 2×2. Уравнение пятиугольника приобретает вид

$$(F_c)_a^d (F_a)_b^c = \sum_e (F_d)_e^c (F_e)_b^d (F_b)_a^c. \tag{8.121}$$

Общее решение для $\mathbf{F} \equiv \mathbf{F}_1$:

$$\mathbf{F} = \begin{pmatrix} \tau & e^{i\phi}\sqrt{\tau} \\ e^{-i\phi}\sqrt{\tau} & -\tau \end{pmatrix}, \tag{8.122}$$

где $e^{i\phi}$ — произвольный фазовый множитель (который при подходящем соглашении о фазах можно положить равным единице), а $\tau = (\sqrt{5}-1)/2 \approx \approx 0.618$, что удовлетворяет уравнению

$$\tau^2 + \tau = 1. \tag{8.123}$$

2×2-матрица $\mathbf{R}$, описывающая перестановку против часовой стрелки двух анионов Фибоначчи, имеет два собственных значения: R^0 для случая, когда полный заряд анионной пары тривиален, и R^1 в случае нетривиального заряда. Уравнение шестиугольника принимает вид

$$R^c(F)^c_a R^a = (F)^c_0 R^0 (F)^0_a + (F)^c_1 R^1 (F)^1_a. \tag{8.124}$$

Используя выражение для $\mathbf{F}$, полученное при решении уравнения пятиугольника, можно решить уравнение шестиугольника для $\mathbf{R}$ и найти

$$\mathbf{R} = \begin{pmatrix} e^{4\pi i/5} & 0 \\ 0 & -e^{2\pi i/5} \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \tau & \sqrt{\tau} \\ \sqrt{\tau} & -\tau \end{pmatrix}. \tag{8.125}$$

Единственным альтернативным решением является комплексно-сопряженное полученному; это второе решение на самом деле описывает ту же модель с точностью до выбора направления обхода в процессе сплетения: по или против часовой стрелки. Следовательно, анионная модель с правилом композиции Фибоначчи *действительно* существует и по сути она единственна.

8.17. Моделирование квантовой схемы с анионами Фибоначчи

Теперь мы знаем достаточно, чтобы задуматься о том, возможно ли моделирование универсального квантового компьютера с помощью анионов Фибоначчи. Необходимо объяснить, как с помощью анионов можно закодировать кубиты и как можно реализовать универсальный набор квантовых вентилей.

Сначала отметим, что гильбертово пространство $V_4^0 \equiv V_{1111}^0$ имеет размерность $N_4^0 = 2$; следовательно, кубит можно закодировать четырьмя анионами с тривиальным полным зарядом. Анионы выстроены в ряд в порядке 1234, пронумерованные слева направо; в стандартном базисном состоянии $|0\rangle$ анионы 1 и 2 объединяются и дают в результате полный заряд 0, тогда как в стандартном базисе $|1\rangle$ анионы 1 и 2 при объединении дают полный заряд 1. Действующий в этом стандартном базисе генератор

группы кос σ_1 (перестановка частиц 1 и 2 против часовой стрелки) представляется матрицей

$$\sigma_1 \longmapsto \mathbf{R} = \begin{pmatrix} e^{4\pi i/5} & 0 \\ 0 & -e^{2\pi i/5} \end{pmatrix}, \tag{8.126}$$

тогда как генератор σ_2 представляется матрицей

$$\sigma_2 \longmapsto \mathbf{B} = \mathbf{F}^{-1}\mathbf{R}\mathbf{F}, \quad \mathbf{F} = \begin{pmatrix} \tau & \sqrt{\tau} \\ \sqrt{\tau} & -\tau \end{pmatrix}. \tag{8.127}$$

Эти матрицы генерируют представление группы кос B_3 на трех нитях, имеющее плотный в SU(2) образ. Действительно, матрицы $\mathbf{R}$ и $\mathbf{B}$ генерируют Z_{10} — подгруппы группы U(2) вокруг двух различных осей, и не существует конечной подгруппы группы U(2), содержащей обе эти подгруппы; следовательно, представление является плотным на группе, содержащей все элементы группы U(2) с детерминантом, равным корню десятой степени из единицы. Аналогично, для n анионов с тривиальным полным зарядом образ представлений группы кос является плотным в $SU(N_n^0)$.

Чтобы смоделировать квантовую схему, действующую на n кубитов, всего используется $4n$ анионов. Мы уже видели, что путем сплетения внутри каждого кластера из четырех анионов можно реализовать произвольные однокубитовые вентили. Чтобы укомплектовать универсальный набор, нам еще потребуются двухкубитовые вентили. Но два соседних кубита кодируются восемью анионами, а перестановки этих анионов генерируют представление группы B_8, образ которого является плотным в $\mathrm{SU}(N_8^0) = \mathrm{SU}(13)$, что, конечно, включает группу SU(4), действующую на два закодированных кубита. Следовательно, каждый вентиль из универсального набора может быть сколь угодно точно смоделирован некоторой конечной косой.

Поскольку мы можем плести косы как по часовой стрелке, так и против, мы также имеем в своем распоряжении обратное значение каждого вентиля перестановок. Следовательно, можно применить теорему Соловея–Китаева и сделать вывод о том, что при длине кос $\mathrm{poly}(\log(1/\varepsilon))$ универсальные вентили модели квантовых схем можно моделировать с точностью ε. Отсюда следует, что идеальная квантовая схема с L вентилями, действующими на все n кубитов сразу, может быть смоделирована с установленной точностью с помощью $4n$ анионов и косы длины $O(L \cdot \mathrm{poly}(\log(L)))$. Как и требовалось, мы показали, что универсальный квантовый компьютер можно эффективно моделировать с помощью анионов Фибоначчи. Отметим, что, в отличие от моделирования с использованием модели неабелева сверхпроводника, для реализации универсальных вентилей не нужны никакие промежуточные измерения.

В вышеприведенном анализе мы предположили полное отсутствие ошибок моделирования, за исключением тех, что ограничивают точность приближения Соловея–Китаева к идеальным вентилям. Следовательно, подразумевается, что температура достаточно низкая по сравнению с энергетической щелью модели, что термически возбужденные анионы слишком редки, чтобы вызвать возмущение, что анионы удерживаются на достаточном расстоянии друг от друга и можно пренебречь неконтролируемой перестановкой зарядов и, наконец, что в топологических квантовых вычислениях ошибки вообще несущественны. Если частота появления ошибок достаточно мала, но не настолько, чтобы ею можно было пренебречь, то для поддержания необходимой точности моделирования можно обратиться к стандартным методам теории квантовой отказоустойчивости, требующим дополнительных полилогарифмических по L накладных расходов. Отказоустойчивая процедура должна включать в себя метод контролирования «утечки» закодированных кубитов, то есть предотвращать уход четырехкубитовых кластеров из двумерного вычислительного пространства V_4^0 в его трехмерное ортогональное дополнительное пространство V_4^1.

8.18. Заключение

Здесь я вкратце коснусь нескольких других тем, которые я мог бы осветить, если бы не кончилось время.

8.18.1. Теория Черна–Саймонса

Мы уже обсуждали, как можно конструировать анионные модели путем прямолинейного решения полиномиальных уравнений. Этот метод прост, но на практике модели часто конструируются с использованием других, более эффективных методов. Действительно, большинство известных анионных моделей были открыты как частные случаи *теории Черна–Саймонса*.

Правила композиции теории Черна–Саймонса представляют собой усеченную версию правил композиции для неприводимых представлений группы Ли. Например, существует ассоциированное с группой SU(2) множество теорий Черна–Саймонса, нумеруемых положительными целыми числами k. Для SU(2) неприводимые представления имеют метки $j = 0, 1/2, 1, 3/2, 2, 5/2, \ldots$, а правила композиции имеют вид

$$j_1 \times j_2 = \sum_{j=|j_2-j_1|}^{j_1+j_2} j. \qquad (8.128)$$

В теории Черна–Саймонса, обозначенной $\mathrm{SU}(2)_k$, полуцелые метки ограничены величиной $j \leqslant k/2$, а метка j содержится в $j_1 \times j_2$, если только $j_1 + j_2 + j \leqslant k$.

Например, модель группы $\mathrm{SU}(2)_1$ — абелева, а нетривиальные правила композиции модели группы $\mathrm{SU}(2)_2$ следующие:

$$\begin{aligned} \tfrac{1}{2} \times \tfrac{1}{2} &= 0 + 1, \\ \tfrac{1}{2} \times 1 &= \tfrac{1}{2}, \\ 1 \times 1 &= 0. \end{aligned} \tag{8.129}$$

Следовательно, метка $1/2$ имеет квантовую размерность $d_{1/2} = \sqrt{2}$, а топологическое гильбертово пространство $2m$ таких анионов с полным зарядом 0 имеет размерность

$$N^0_{(\frac{1}{2})^{2m}} = 2^{m-1}. \tag{8.130}$$

Полиномиальные уравнения для этих правил композиции имеют множество решений (из которых лишь одно описывает свойства сплетения модели группы $\mathrm{SU}(2)_2$), но ни одна из результирующих моделей не имеет правил сплетения, универсальных с точки зрения реализации квантовых вычислений. Пространство $V^0_{\frac{1}{2}\frac{1}{2}\frac{1}{2}\frac{1}{2}}$ двумерно, а 2×2-матрицы $\mathbf{F} \equiv \mathbf{F}_{\frac{1}{2}\frac{1}{2}\frac{1}{2}\frac{1}{2}}$ и $\mathbf{R} \equiv \mathbf{R}_{\frac{1}{2}\frac{1}{2}}$, с точностью до общих фаз и комплексного сопряжения, имеют вид

$$\mathbf{F} = \mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \mathbf{R} = \mathbf{P} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{8.131}$$

Это квантовые вентили группы Клиффорда, не отвечающие требованиям универсальности.

Однако при $k \geqslant 3$ $\mathrm{SU}(2)_k$-модели *вычислительно универсальны*. Нетривиальные правила композиции группы $\mathrm{SU}(2)_3$ имеют вид

$$\begin{aligned} \tfrac{1}{2} \times \tfrac{1}{2} &= 0 + 1, \\ \tfrac{1}{2} \times 1 &= \tfrac{1}{2} + \tfrac{3}{2}, \\ \tfrac{1}{2} \times \tfrac{3}{2} &= 1, \end{aligned}$$

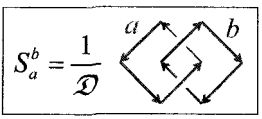
$$
\begin{aligned}
1 \times 1 &= 0 + 1, \\
1 \times \frac{3}{2} &= \frac{1}{2}, \\
\frac{3}{2} \times \frac{3}{2} &= 0.
\end{aligned}
\tag{8.132}
$$

Рассмотренная нами модель Фибоначчи (Янга–Ли) получена путем усечения группы $\mathrm{SU}(2)_3$, состоящем в исключении нецелых меток $1/2$ и $3/2$ (то есть, это теория Черна–Саймонса $\mathrm{SO}(3)_3$); тогда единственным остающимся нетривиальным правилом композиции является $1 \times 1 = 0 + 1$.

Уонг (неопубликовано) недавно сконструировал все анионные модели с количеством меток не более четырех и обнаружил, что все они тесно связаны с открытыми теорией Черна–Саймонса моделями.

8.18.2. S-матрица

Модулярную S-матрицу анионной модели можно определить на языке мировых линий двух анионов, образующих *зацепление Хопфа*:



Здесь $\mathcal{D}$ — полная квантовая размерность модели, и мы воспользовались нормировкой, в которой расцепленные петли имели бы значение $d_a d_b$; тогда матрица S_a^b симметрична и унитарна. В абелевых анионных моделях понятие зацепления Хопфа возникло при обсуждении топологического вырождения, которым определялось воздействие на вакуумное состояние анионной модели, возникающее при обходе аниона вокруг одного из циклов тора. В неабелевом случае S-матрица имеет аналогичную интерпретацию. С помощью элементарных рассуждений S может быть связана с правилами композиции:

$$
(N_a)_b^c = \sum_d S_b^d \left(\frac{S_a^d}{S_1^d} \right) (S^{-1})_d^c; \tag{8.133}
$$

то есть S-матрица одновременно диагонализует все матрицы $\{N_a\}$ (соотношение Верлинде). Отметим, что из определения следует, что $S_1^a = d_a/\mathcal{D}$.

8.18.3. Краевые возбуждения

В нашей формулировке анионных моделей мы обсудили объединение и сплетение частиц в двумерном *объеме*. Но существует другой аспект фи-

зики двумерных сред, который до сих пор не обсуждался, — свойства одномерной *границы* образца. Обычно, если двумерная система содержит анионы в объеме, то имеются и *киральные безмассовые возбуждения*, распространяющиеся вдоль одномерной границы. При ненулевой температуре T вдоль границы существует поток энергии, определяемый выражением

$$J = \frac{\pi}{12} c_- T^2; \tag{8.134}$$

здесь постоянная c_-, называемая *киральным центральным зарядом* границы, является универсальным свойством, неуязвимым для небольших изменений в гамильтониане системы.

В то время как этот киральный центральный заряд является внутренним свойством двумерной среды, свойства анионов в объеме не определяют его полностью; скорее, мы имеем

$$\frac{1}{\mathcal{D}} \sum_a d_a^2 e^{2\pi i J_a} = e^{(2\pi i/8)c_-}, \tag{8.135}$$

где суммирование ведется по полному набору меток анионной модели, а $e^{2\pi i J_a} = R^1_{a\bar{a}}$ — топологический спин метки a. Это выражение связывает величину c_-, характеристику краевой теории, с квантовыми размерностями и топологическими спинами объемной теории, но определяет c_- только по модулю восемь. Следовательно, по крайней мере, в принципе, возможно существование множества краевых теорий, соответствующих единственной теории анионов в объеме.

8.19. Библиографические замечания

Некоторые из пионерских работ по теории анионов напечатаны в [1].

Описанная мной модель «неабелева сверхпроводника» в литературе часто упоминается как «квантовый дубль», она изучается с использованием теории представлений алгебры Хопфа (см., например, [2]).

В [3] было впервые предложено использование неабелевых анионов для отказоустойчивых квантовых вычислений. В этой работе также обсуждается торический код и связанные с ним решеточные модели, имеющие неабелевы фазы. Конкретная реализация универсальных квантовых вычислений в неабелевом сверхпроводнике была рассмотрена в [4, 5]. Мое обсуждение набора универсальных вентилей опирается на [6], где также рассматриваются более общие модели. Другие схемы, которые обеспечивают более экстенсивное использование электрических зарядов и являются универсальными для меньших групп (таких как S_3), описаны в [7].

Графические методы, подобные тем, что я использовал в обсуждении квантовой размерности, широко используются для вывода свойств анионов в [8]. Роль полиномиальных уравнений (уравнений пяти- и шестиугольника) в (1+1)-мерной конформной теории поля обсуждается в [9].

Моделирование анионов с использованием квантовой схемы описано в [10]. Моделирование универсального квантового компьютера с использованием анионов $SU(2)_{k=3}$-теории Черна–Саймонса обсуждается в [11]. В [12] было отмечено, что модель Янга–Ли также является универсальной.

В моих лекциях я не обсуждал физическую реализацию, но в любом случае здесь я перечислю несколько относящихся к данной проблеме работ. Идеи реализации абелевых и неабелевых анионов с помощью массивов сверхпроводящих джозефсоновских контактов описаны в [13]. Спиновая модель со взаимодействием между ближайшими соседями, которая имеет неабелевы анионы (хотя не только они являются вычислительно универсальными), предложена и решена в [14], а предложение реализовать эту модель с использованием холодных атомов, захваченных в оптической решетке, описано в [15]. Некоторые идеи реализации (вычислительно универсальной) модели $SU(2)_{k=3}$ в системе взаимодействующих электронов обсуждаются в [16].

Своим пониманием теории квантовых вычислений с помощью неабелевых анионов я обязан многочисленным полезным дискуссиям с Алексеем Китаевым.

Литература

[1] F. Wilczek, *Fractional statistics and anyon superconductivity* (World Scientific, Singapore, 1990).

[2] M. de Wild Propitius and F. A. Bais, Discrete gauge theories, arXiv: hep-th/9511201 (1995); CRM-CAP Summer School «Particles and Fields 94», Bannf, Alberta, Canada, August 16–24, (1994).

[3] A. Yu. Kitaev, Fault-tolerant quantum computation by anyons, Annals Phys. **303**, 2–30 (2003), arXiv: quant-ph/9707021.

[4] R. W. Ogburn and H. Preskill, Topological quantum computation, in *Lect. Notes in Comp. Sci.* C.P. Williams (ed.) **1509**, 341–356, Springer–Verlag (1999).

[5] J. Preskill, Fault-tolerant quantum computation, arXiv: quant-ph/9712048 (1997); В книге: *Introduction to Quantum Computation*, H.-K. Lo, S. Popesku, T.P. Spiller (Eds.), World Scientific, Singapore et al. (1998); пере-

вод: Дж. Прескилл, Отказоустойчивые квантовые вычисления (в этом издании).

[6] C. Mochon, Anyons from non-solvable finite groups are sufficient for universal quantum computation Phys. Rev. A **67**, 022315 (2003), quant-ph/0206128.

[7] C. Mochon, Anyon computers with smaller groups, Phys. Rev. A **69**, 032306 (2004), arXiv: quant-ph/0306063.

[8] J. Fröhlich and F. Gabbiani, Braid statistics in local quantum theory, Rev. Math. Phys. **2:3**, 251–353 (1990).

[9] G. Moore and N. Seiberg, Classical and quantum conformal field theory, Comm. Math. Phys. **123**, 171–254 (1989).

[10] M. H. Freedman, A. Kitaev, and Z. Wang, Simulation of topological field theories by quantum computers, Comm. Math. Phys. **227**, 587–603 (2002), arXiv: quant-ph/0001071.

[11] M. H. Freedman, M. Larsen, and Z. Wang, A modular functor which is universal for quantum computation, Comm. Math. Phys. **227**, 605–622 (2002), arXiv: quant-ph/0001108 (2000).

[12] G. Kuperberg, unpublished.

[13] B. Doucot, L. B. Iofie, and J. Vidal, Discrete non-Abelian gauge theories in two-dimensional lattices and their realizations in Josephson-junction arrays, Phys. Rev. B **69**, 214501 (2004), arXiv: cond-mat/0302104.

[14] A. Kitaev, Anyons in a spin model on the honeycomb lattice, unpublished. [Обсуждение точно решаемой спиновой модели Китаева на решетке «пчелиных сот» см. в: A. Kitaev, C. Laumann, Topological phases and quantum computation, cond-mat/0904.2771; A. Kitaev, Anyons in an exactly solved model and beyond, *Ann. Phys.*, **321**, 2–111 (2006); cond-mat/0506438]

[15] L.-M. Duan, E. Demler, and M. D. Lukin, Controlling spin exchange interactions of ultracold atoms in optical lattices, Phys. Rev. Lett. **91**, 090402 (2003), arXiv: cond-mat/0210564.

[16] M. Freedman, C. Nayak, K. Shtengel, K. Walker, and Zhenghan Wang, A class of P; T-invariant topological phases of interacting electrons, *Ann. Phys.*, NY, **310**, 428–492; cond-mat/0307511 (2003).

Приложение

Отказоустойчивые квантовые вычисления

Надежные квантовые компьютеры[1]

Джон Прескилл[2]

Новая область науки *коррекция квантовых ошибок* получила заметное развитие с момента своего возникновения менее двух лет назад. Закодированную квантовую информацию можно защитить от ошибок, возникающих вследствие неконтролируемых взаимодействий с окружающей средой. Исправление может эффективно выполняться даже при возникновении случайных ошибок во время самой процедуры восстановления. Более того, закодированная квантовая информация может *обрабатываться* без существенного размножения ошибок. Следовательно, квантовые вычисления произвольной продолжительности могут надежно выполняться при условии, что средняя вероятность одной ошибки на квантовый вентиль меньше некоторого критического значения, называемого *порогом безошибочности*. Квантовый компьютер, вмещающий порядка 10^6 кубитов информации, при вероятности ошибки на квантовый вентиль порядка 10^{-6}, мог бы стать внушительной вычислительной машиной. Даже меньший и менее точный квантовый компьютер смог бы выполнить множество полезных задач.

Данная работа основана на материалах доклада, представленного на Конференции по квантовой когерентности и декогерентизации, Институт теоретической физики, Санта Барбара, 15–18 декабря 1996 г.

1. Золотой век коррекции квантовых ошибок

Многие из нас рассчитывают на то, что в XXI веке квантовые компьютеры, наконец, станут практичными и полезными вычислительными

[1] J. Preskill, Reliable Quantum Computers, *Proc. Roy. Soc. London* A, **454**, pp. 385–410 (1998).

[2] Калифорнийский технологический институт, Пасадена, CA 91125, США.

устройствами. Однако справедливости ради следует отметить, что сейчас никто из нас не может точно предсказать, что будет представлять собой «железо» этих машин будущего; возможно, оно будет значительно отличаться от аппаратных средств, исследуемых в наши дни физиками-экспериментаторами. Но в одном можно быть вполне уверенным — работа реального квантового компьютера будет включать в себя определенный тип коррекции ошибок. Квантовые компьютеры гораздо более восприимчивы к возникновению ошибок по сравнению с традиционными цифровыми компьютерами, поэтому для предотвращения сбоев в их работе потребуются определенные методы контроля и исправления этих ошибок.

Еще в середине 90-х годов мы не имели четкого представления о том, как будет работать коррекция квантовых ошибок и будет ли она работать вообще. Действительно, был ряд причин для пессимизма относительно реальной возможности исправления квантовых ошибок (Unruh 1995; Landauer 1995, 1996, 1997). Во-первых, несмотря на то, что были разработаны достаточно изощренные методы исправления ошибок в классической информации (MacWilliams & Sloane 1977), было далеко не очевидно, как их адаптировать для исправления атакующих квантовые системы *фазовых* ошибок. Во-вторых, в квантовом компьютере, как и в классическом аналоговом компьютере, с течением времени малые ошибки могут накапливаться и, в конечном счете, превращаться в большие, а найти методы, способные предотвращать или исправлять подобные малые ошибки, достаточно сложно. В-третьих, чтобы исправить ошибку, необходимо сначала получить определенную информацию о ее природе, осуществив измерение, которое влечет за собой опасность повреждения закодированной в устройстве чувствительной квантовой информации. Наконец, чтобы защититься от ошибок, необходимо кодировать информацию в избыточном виде. Однако известная теорема (Wootters & Zurek 1982; Dieks 1982) говорит о том, что квантовую информацию нельзя клонировать и, следовательно, не ясно, как хранить (квантовую) информацию с требуемой избыточностью.

Но к настоящему моменту все эти трудности преодолены — теперь мы знаем, что коррекция квантовых ошибок действительно возможна. Ключевая идея, осознанная нами, состоит в том, что с *запутыванием можно бороться с помощью запутывания*. Запутанность квантовых состояний может быть нашим врагом, поскольку запутывание состояния нашего устройства с окружающей средой может скрывать от нас квантовую информацию и, таким образом, служить причиной ошибок. Но запутанность может быть и нашим союзником — информацию, которую мы хотим защитить, можно закодировать в запутанном состоянии, то есть в корреляциях, охватывающих большое количество кубитов. Тогда эту информацию невозможно из-

влечь, измеряя лишь несколько кубитов. Впрочем, по той же причине она не может быть и повреждена, если окружающая среда взаимодействует лишь с несколькими кубитами. Более того, мы узнали, что хотя квантовый компьютер является до известной степени аналоговым устройством, совершаемые им ошибки можно *оцифровать*. Мы боремся с малыми ошибками, выполняя подходящие измерения, которые проецируют состояние квантового компьютера или на исходное неповрежденное состояние, или на состояние с большой ошибкой, которую можно исправить известными методами. Мы осознали, что возможно измерение ошибок без измерения самой информации — можно получить информацию об истинной природе ошибки, ничего при этом не узнав о закодированной в нашем устройстве квантовой информации (что могло бы повлечь за собой декогерентизацию и сбой в нашем вычислении). Основная идея коррекции квантовых ошибок состоит в следующем: маленькое подпространство гильбертова пространства нашего устройства определяется как *кодовое подпространство*. Оно выбирается таким образом, чтобы любая ошибка, которую мы хотим исправить, перемещала его в одно из взаимно ортогональных *подпространств ошибок*. После того как наша система провзаимодействовала с окружением, выполняется измерение, результат которого сообщает о том, в каком из этих взаимно ортогональных подпространств находится система, и, следовательно, точно определяет тип возникшей ошибки. После этого ошибку можно исправить, применяя подходящее унитарное преобразование.

Если мы и были скептически настроены в 1995 году, то к концу 1996 года появилась веская причина для оптимизма. Этот год стал поворотным пунктом в истории квантовой информации; это момент, когда мы узнали, как противостоять и обращать эффекты декогерентизации. Это открытие имеет важное значение для квантовых вычислений, но на самом деле его последствия гораздо шире. Вот несколько основных этапов, пройденных в этом году:

- Осенью 1995 года Питер Шор (Shor 1995) и Эндрю Стин (Steane 1996a) впервые указали на то, что коды, корректирующие квантовые ошибки, существуют.

- К началу 1996 года Стин (Steane 1996b), а также Колдербэнк и Шор (Calderbank & Shor 1996) показали существование *хороших* кодов, то есть кодов, способных корректировать большое количество ошибок.

- Из работы по случайным кодам Ллойда (Lloyd 1997) и Беннетта, Дивинченцо, Смолина и Вутерса (Bennett, *et al.* 1996) мы узнали, что

квантовая информация может храниться в течение достаточно длительного времени. То есть существует код, который позволяет с высокой точностью вопроизведения восстанавливать хранящуюся информацию при условии, что вероятность ошибки на кубит ориентировочно менее 19%.

Но эта оценка допустимой частоты ошибок (19%) довольно обманчива, поскольку она применима только при весьма нереалистичном допущении, согласно которому кодирование информации и ее восстановление выполняются безупречно, то есть без каких-либо ошибок. В действительности же кодирование и восстановление сами по себе являются сложными квантовыми вычислениями, и в процессе их выполнения неизбежно появление ошибок. Таким образом, нам необходимо найти достаточно устойчивые методы избавления от ошибок, чтобы можно было по-прежнему с высокой точностью восстанавливать квантовую информацию, даже если ошибки совершаются в самом процессе восстановления. Это задача *отказоустойчивого восстановления*, и Питер Шор (Shor 1996) показал в своей пионерской работе, написанной в мае 1996 года, что оно возможно, если частота возникновения ошибок не слишком велика.[1] Конечно, нам необходимо больше, чем просто хранение информации; мы хотим иметь возможность обрабатывать данную информацию и выполнять интересные квантовые вычисления. Поэтому мы должны показать возможность разработки квантовых вентилей, эффективно работающих с квантовой информацией, закодированной так, чтобы быть защищенной от ошибок. В той же работе Шор показал, что отказоустойчивые *вычисления* действительно возможны.

В августе того же года Мэнни Нилл и Раймонд Лафламм (Knill & Laflamme 1996) показали существование порога безошибочности хранения квантовой информации: если частота появления ошибок ниже некоторого критического значения, то возможно сколь угодно длительное хранение неизвестного квантового состояния с высокой точностью воспроизведения. В калтеховской группе (Gottesman *et al.* 1996) мы быстро поняли, что можно объединить идеи Шора и Нилла и Лафламма и показать, что порог безошибочности существует и для вычислений; если частота появления ошибок ниже некоторого критического значения, то возможно выполнение сколь угодно продолжительного квантового вычисления при ничтожной вероятности возникновения ошибок. Аналогичные выводы были сделаны Нилом, Лафламмом и Зуреком (Knill *et al.* 1996, 1997), Аароновым и Бен-Ором (Aharonov & Ben-Or 1996), а также Китаевым (Kitaev 1996b).

[1] Методы отказоустойчивого восстановления независимо развивались Алексеем Китаевым (Kitaev 1996a).

Итак, мы столкнулись с острой проблемой разработки и создания квантового компьютера. Мы можем лишь сказать, насколько хорошо должно работать аппаратное обеспечение этой машины, чтобы оно могло быть использовано для выполнения интересных квантовых вычислений, конкурентноспособных с теми, что могут выполнять лучшие современные цифровые компьютеры. Начнем с обзора основных принципов отказоустойчивых вычислений.

2. Законы отказоустойчивых вычислений

Чтобы выполнять отказоустойчивые вычисления, необходимо: (1) обеспечить *надежное* исправление ошибок, (2) уметь применять вентили, способные *обрабатывать* закодированную информацию, (3) контролировать распространение ошибок. Когда вентиль применяется, скажем, к паре кубитов, один из которых содержит ошибку, то эта ошибка будет стремиться перейти на другой кубит. Необходимо соблюдать осторожность, чтобы сдерживать заражение. Правила, которым мы должны следовать при выполнении отказоустойчивых вычислений, можно систематизировать, как я это буду называть, в виде «законов отказоустойчивых вычислений». Эту информацию можно получить из пионерской работы Шора (Shor 1996).

Первый закон гласит: **(1) Не используйте один и тот же кубит дважды.**[1] Плохая сеть XOR-вентилей, нарушающая эту заповедь, изображена на рис. 2 (используются условные обозначения рис. 1). Бит, который я назвал служебным, является целью нескольких следующих друг за другом вентилей. Если в этом служебном бите есть хотя бы одна ошибка, такая сеть может распространить ее на некоторые другие биты, являющиеся источниками XOR-вентилей; следовательно, инфекция распространяется лавинообразно. Странная квантово-механическая особенность состоит в следующем: тогда как даже классический XOR-вентиль распространяет ошибки инвертирования бита от источника к цели, в случае квантовых вентилей следует помнить о фазовых ошибках, а они распространяются в противоположном направлении от цели к источнику. Итак, при выполнении квантовых вычислений необходимо быть особенно внимательным к возможности распространения ошибок. Следуя этому закону, сеть можно перестроить, как это показано на том же рисунке 2, увеличив количество служебных кубитов до нескольких, так чтобы ни один из них не действовал более одного раза.

Второй закон касается того, как должно выполняться исправление ошибок. С этой целью некоторая информация из блока данных копиру-

[1]Менее строгая, но более благоразумная версия этого закона: избегайте использования одного и того же кубита слишком часто.

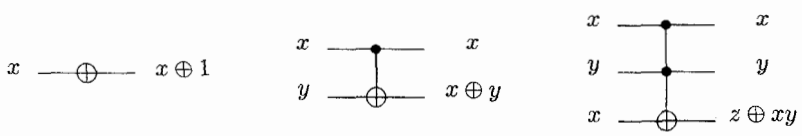


Рис. 1. Графическое изображение вентилей NOT, XOR (исключающее ИЛИ) и вентиля Тоффоли (дважды контролируемое НЕ)

Рис. 2. Первый закон. Плохая схема, которая несколько раз использует один и тот же служебный кубит, и хорошая схема, которая использует каждый вспомогательный бит только один раз

ется в блок служебных кубитов. Затем выполняется его измерение, чтобы найти *синдром ошибки*, который сообщает, какая операция восстановления необходима. Второй закон гласит: **(2) Копируйте ошибки, а не информацию**. Если какая-то часть закодированной информации копируется из блока данных в служебный, то результирующее запутывание между этими регистрами вызовет декогерентизацию и, следовательно, ошибки. Чтобы избежать этого, необходимо перед копированием какой-либо информации подготовить служебные кубиты в специальных состояниях, выбранных таким образом, чтобы результаты их измерения сообщали информацию лишь о возникших ошибках, а не о закодированных данных (см. рис. 3).

Третий и четвертый законы требуют: прежде чем что-либо делать, необходимо убедиться (если это возможно) в том, что это делается правильно. Часто требуется приготовить блок, кодирующий известное квантовое состояние, такое как закодированный нуль. Третий закон гласит: **(3) Кодируя *известное* квантовое состояние, *проверяйте* результат**. В процессе кодирования информация оказывается особенно подверженной действию ошибок: защитные механизмы кода еще не работают и одна-единственная ошибка может распространиться катастрофическим образом. Следователь-

Рис. 3. Второй закон. Служебный кубит должен быть подходящим образом приготовлен

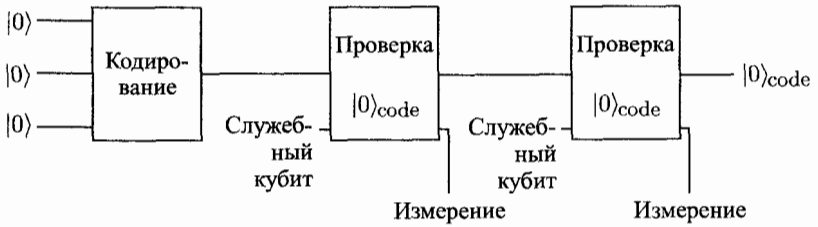


Рис. 4. Третий закон. Кодирующее устройство конструирует $|0\rangle_{\text{code}}$, а затем выполняется неразрушающее измерение (по крайней мере дважды), чтобы убедиться в успешности кодирования

но, необходимо выполнить измерение, проверяющее, правильность кодирования. Конечно, и сама проверка может быть ошибочной, поэтому ее следует повторить несколько раз, пока мы в достаточной мере не убедимся в правильности кодирования.

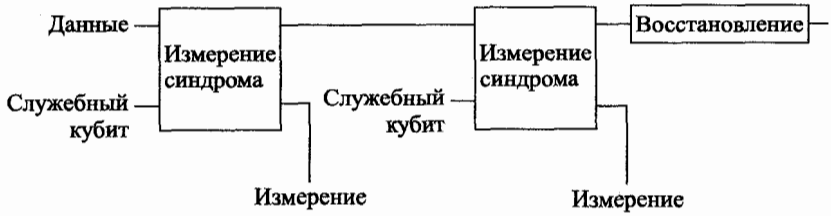


Рис. 5. Четвертый закон. Для обеспечения точности необходимо повторять операции, в частности, измерение синдрома

Необходимость повторения проверок фактически представляет собой частный случай четвертого закона, который гласит: **(4) *Повторяйте* опе-**

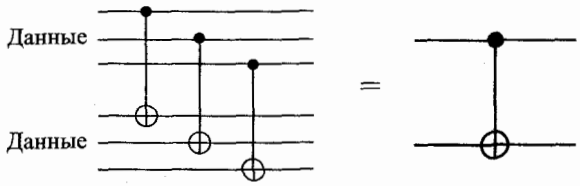


Рис. 6. Пятый закон. Трансверсальная (побитовая) реализация отказоустойчивого XOR-вентиля

рации. Этот закон важно применять к измерению синдрома, предшествующему процедуре исправления. Ошибка, появившаяся в процессе измерения синдрома, может *одновременно повредить данные и явиться причиной* ошибочного синдрома. Если мы по ошибке примем измеренный синдром и будем действовать в соответствии с ним, то вместо исправления это приведет к дальнейшему повреждению данных. Следовательно, перед выполнением восстановления необходимо быть в высшей степени уверенными в том, что измерение синдрома выполнено корректно. Для достижения достаточной уверенности необходимо, в соответствии с четвертым законом, повторить измерение синдрома несколько раз.

Пятый закон является, вероятно, наиболее важным и в то же время наиболее сложным для выполнения: **(5) Используйте *правильный* код.** Используемый для вычислений код должен обладать специальными качествами: допускать применение квантовых вентилей к закодированной информации, эффективно работать и отвечать первым четырем законам. Например, хороший для вычислений код может быть таким, чтобы действующий на закодированные кубиты XOR-вентиль применялся, как показано на рисунке 6: по одному XOR-вентилю, применяемому к каждому кубиту, как в блоке источников, так и в блоке целей. Тогда действующий на закодированные кубиты вентиль удовлетворяет первому закону и является отказоустойчивым.[1]

3. Пример: 7-кубитовый код Стина

Чтобы понять, как осуществляется коррекция квантовых ошибок, поучительно рассмотреть конкретный код. Простым и важным примером ко-

[1]Когда я читал лекцию в декабре, принцип применения отказоустойчивых вентилей для большинства квантовых кодов еще не был известен. Позднее Готтесман (Gottesman 1997a) показал, что отказоустойчивые вычисления возможны для всех «стабилизирующих кодов». Тем не менее, пятый закон — хороший закон; подойдет любой код, но для некоторых из них реализация отказоустойчивых вентилей проще.

да, корректирующего квантовые ошибки, является разработанный Эндрю Стином (Steane 1996ab) 7-кубитовый код. Он позволяет хранить один кубит квантовой информации (произвольное состояние в двумерном гильбертовом пространстве), в совокупности используя семь кубитов (погружая двумерное гильбертово пространство в 2^7-мерное пространство). Фактически код Стина тесно связан с известным нам классическим корректирующим ошибки [7,4,3]-кодом Хэмминга (MacWilliams & Sloane 1977). Чтобы понять, как работает код Стина, для начала полезно разобраться в устройстве кода Хэмминга.

Код Хэмминга использует блок из семи битов для кодирования четырех битов классической информации: имеется $16 = 2^4$ строк длины семь, которые представляют собой кодовые слова, характеризуемые с помощью матрицы контроля четности,

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{1}$$

Каждое кодовое слово является 7-битовой строкой v_{code}, удовлетворяющей уравнению

$$\sum_k H_{jk}(v_{\text{code}})_k = 0 \pmod 2; \tag{2}$$

то есть в арифметике по модулю 2 матрица H уничтожает каждое кодовое слово. Поскольку $Z_2 = \{0, 1\}$ представляет собой (конечное) поле, на нем применимы известные результаты линейной алгебры. Матрица H имеет три линейно независимых строки, а ее ядро[1] является линейной оболочкой четырех независимых векторов-столбцов. 16 кодовых слов представляют собой все возможные линейные комбинации этих четырех строк с коэффициентами, выбираемыми из $\{0, 1\}$.

Теперь предположим, что v_{code} — (неизвестное) кодовое слово, в котором возникла единственная (неизвестная) ошибка: один из семи битов инвертировался. Наша задача — определить поврежденный бит и исправить ошибку. Этот трюк можно выполнить с помощью матрицы контроля четности. Пусть e_i представляет собой строку с единицей в i-й позиции и нулями в остальных. Тогда при инвертировании i-го бита v_{code} становится равным $v_{\text{code}} + e_i$. Если мы подействуем на эту строку матрицей H, то получим

$$H(v_{\text{code}} + e_i) = He_i \tag{3}$$

[1] Ядро матрицы или ее *нуль-пространство* — пространство векторов, удовлетворяющих уравнению (2). — *Прим. ред.*

(поскольку H уничтожает v_{code}), что представляет собой именно i-й столбец матрицы H. А поскольку все столбцы матрицы H различны, то этим однозначно определяется значение i. Выяснив местоположение ошибки, ее можно исправить, инвертируя i-й бит в исходное состояние. Таким образом, если инвертируется только один бит, то можно однозначно восстановить закодированную информацию; но если происходит инвертирование двух или более различных битов, закодированная информация будет повреждена. Замечательно здесь то, что величина He_i выявляет позицию ошибки, ничего не сообщая о v_{code}, то есть не раскрывая закодированную информацию.

Код Стина представляет собой квантовое обобщение этого классического кода коррекции ошибок. Он использует 7-кубитовый «блок» для кодирования одного кубита квантовой информации, то есть произвольного состояния в двумерном гильбертовом пространстве, натянутом на два состояния: «логический нуль» $|0\rangle_{\text{code}}$ и «логическая единица» $|1\rangle_{\text{code}}$. Конструкция кода позволяет исправить любую ошибку, возникшую в любом из семи кубитов в блоке.

Что подразумевается под произвольной ошибкой? Вызывающий подозрение кубит мог подвергаться случайному *унитарному* преобразованию, или *потерять когерентность*, запутавшись с состояниями окружающей среды. Пусть неповрежденный кубит находится в состоянии $a|0\rangle + b|1\rangle$. (Конечно, этот отдельный кубит может быть запутан с другими, так что коэффициенты a и b не обязательно являются комплексными числами; они могут представлять собой векторы состояний, ортогональных как $|0\rangle$, так и $|1\rangle$, которые мы (пока) считаем неуязвимыми для ошибок.) Если теперь кубит поражен произвольной ошибкой, результирующее состояние можно разложить следующим образом:

$$\begin{aligned} a|0\rangle + b|1\rangle \;\rightarrow\; & (a|0\rangle + b|1\rangle) \otimes |A_{\text{no error}} \rangle_{\text{env}} + \\ & + (a|1\rangle + b|0\rangle) \otimes |A_{\text{bit-flip}} \rangle_{\text{env}} + \\ & + (a|0\rangle - b|1\rangle) \otimes |A_{\text{phase-flip}} \rangle_{\text{env}} + \\ & + (a|1\rangle - b|0\rangle) \otimes |A_{\text{both errors}} \rangle_{\text{env}}, \end{aligned} \tag{4}$$

где каждое $|A\rangle_{\text{env}}$ обозначает состояние окружающей среды. Мы не делаем никаких особенных предположений относительно ортогональности или нормировки состояний окружения $|A\rangle_{\text{env}}$,[1] поэтому уравнение (4) не приводит ни к какой потере общности. Таким образом, кубит эволюционирует

[1] Хотя, конечно, совместная эволюция кубита и окружающей среды должна быть унитарной.

к линейной суперпозиции четырех возможностей: (1) никакой ошибки не возникает, (2) возникает инвертирование бита $|0\rangle \leftrightarrow |1\rangle$, (3) обращается относительная фаза $|0\rangle$ и $|1\rangle$, (4) инвертирование бита и обращение фазы происходят одновременно.

Теперь понятно, как должен работать квантовый код коррекции ошибок (Steane 1996; Knill & Laflamme 1997). Совершая подходящее измерение, мы хотим диагностировать, какая из этих четырех возможностей случилась на самом деле. Конечно, в целом, состояние кубита будет представлять линейную комбинацию этих четырех состояний, но наше измерение должно спроецировать его на использованный в (4) базис. После этого мы можем приступить к исправлению ошибки с помощью одного из четырех унитарных преобразований:

$$(1)\quad I,\quad (2)\quad X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\quad (3)\quad Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},\quad (4)\quad X \cdot Z, \quad (5)$$

(какое из них необходимо применить, укажет результат измерения). Применив это преобразование, мы возвращаем кубит в его исходное состояние и полностью распутываем квантовые состояния кубита и окружающей среды. Существенно, что при диагностике ошибки мы ничего не узнаем о закодированной информации, поскольку получение любой информации о коэффициентах a и b в уравнении (4) неизбежно разрушит когерентность кубита.

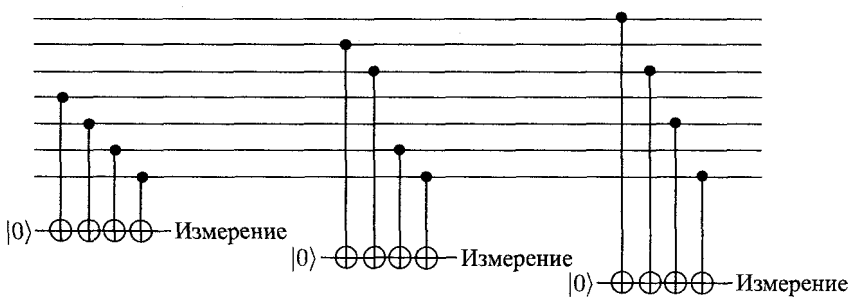


Рис. 7. Вычисление синдрома инвертирования бита для 7-кубитового кода Стина. Повторение вычисления в повернутом базисе диагностирует ошибку обращения фазы. Чтобы сделать процедуру отказоустойчивой, каждый служебный кубит должен быть заменен четырьмя кубитами в подходящих состояниях

Если мы используем код Стина, то удовлетворяющее этим критериям измерение возможно. Логический нуль является равновзвешенной суперпозицией всех кодовых слов Хэмминга (H) с четным весом (с четными количеством единиц),

$$\begin{aligned}|0\rangle_{\text{code}} &= \frac{1}{\sqrt{8}} \sum_{v_{\text{even}} \in \text{H}} |v_{\text{even}}\rangle = \\ &= \frac{1}{\sqrt{8}} \big(|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle + \\ &\quad + |1010101\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle\big),\end{aligned} \tag{6}$$

а логическая единица является равновзвешенной суперпозицией всех кодовых слов Хэмминга с нечетным весом (с нечетным количеством единиц),

$$\begin{aligned}|1\rangle_{\text{code}} &= \frac{1}{\sqrt{8}} \sum_{v_{\text{odd}} \in \text{H}} |v_{\text{odd}}\rangle = \\ &= \frac{1}{\sqrt{8}} \big(|1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle + \\ &\quad + |0101010\rangle + |0100101\rangle + |0011001\rangle + |0010110\rangle\big).\end{aligned} \tag{7}$$

Поскольку все возникающие в уравнениях (6) и (7) состояния являются кодовыми словами Хэмминга, инвертирование единственного бита в блоке несложно детектировать, выполняя простое квантовое вычисление, как это показано на рисунке 7. Мы дополняем блок из семи кубитов тремя служебными кубитами[1] и выполняем унитарное преобразование:

$$|v\rangle \otimes |0\rangle_{\text{anc}} \rightarrow |v\rangle \otimes |Hv\rangle_{\text{anc}}, \tag{8}$$

где H — матрица контроля четности Хэмминга, а $|\cdot\rangle_{\text{anc}}$ обозначает состояние трех служебных кубитов. Если ошибка содержится лишь в одном из семи кубитов, то измерение служебного кубита проецирует его либо на инвертированное состояние, либо на исходное неповрежденное (но не на любую нетривиальную суперпозицию этих двух состояний). В первом случае результат измерения диагностирует поврежденный бит, ничего сообщая о закодированной в блоке квантовой информации.

Но чтобы выполнять коррекцию квантовых ошибок, нам понадобится диагностировать не только ошибки инвертирования битов, но и фазовые ошибки. Для достижения этой цели мы можем, следуя Стину

[1] Чтобы сделать процедуру отказоустойчивой, нам понадобится увеличить количество служебных кубитов, как это обсуждается в разделе 4.

(Steane 1996ab), изменить базис для каждого кубита, применив поворот Адамара

$$R = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{9}$$

Тогда ошибки, бывшие фазовыми в базисе $|0\rangle$, $|1\rangle$, в повернутом базисе

$$|\tilde{0}\rangle \equiv \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big), \quad |\tilde{1}\rangle \equiv \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \tag{10}$$

становятся ошибками инвертирования бита. Следовательно, будет достаточно, если наш код способен диагностировать ошибки инвертирования бита в этом повернутом базисе. Но если мы применяем поворот Адамара к каждому из семи кубитов, то логические нуль и единица кода Стина в новом базисе приобретают вид

$$\begin{aligned} |\tilde{0}\rangle &= \frac{1}{4}\sum_{v\in \mathrm{H}} |v\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big), \\ |\tilde{1}\rangle &= \frac{1}{4}\sum_{v\in \mathrm{H}} (-1)^{\mathrm{wt}(v)} |v\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \end{aligned} \tag{11}$$

(где $\mathrm{wt}(v)$ обозначает вес v). Важно то, что $|\tilde{0}\rangle$ и $|\tilde{1}\rangle$, как и $|0\rangle$ и $|1\rangle$, являются суперпозициями кодовых слов Хэмминга. Следовательно, в повернутом базисе, как и в исходном, мы можем выполнить контроль четности Хэмминга для диагностики инвертирования битов, которое в исходном базисе представляло собой обращение фазы. При условии, что поврежден только один кубит, выполнение контроля четности в обоих базисах полностью диагностирует ошибку и дает возможность ее исправить.

В описанной выше схеме исправления ошибок я предполагал, что ошибка воздействует только на один кубит в блоке. Очевидно, это предположение нереалистично; обычно все кубиты до некоторой степени запутываются с окружением. Но, как мы уже видели, процедура определения синдрома ошибки обычно проецирует каждый кубит на исходное неповрежденное состояние. Для каждого кубита существует ненулевая предполагаемая малой вероятность возникновения ошибки, которую мы обозначим как ϵ. Сейчас мы сделаем очень важное предположение: ошибки, действующие на разные кубиты в одном блоке, абсолютно не коррелируют между собой. При таком допущении вероятность возникновения двух ошибок имеет порядок ϵ^2 и, следовательно, гораздо меньше вероятности возникновения одной ошибки, если ϵ — достаточно малая величина. Поэтому, с точностью

порядка ϵ, мы можем уверенно сосредоточить наше внимание на случае, когда ошибку содержит максимум один кубит в блоке.

Но в случае (маловероятном), когда в одном и том же блоке кода возникает две ошибки, наша процедура восстановления, как правило, будет безуспешной. Если в одном блоке инвертируются два бита, то контроль четности Хэмминга неправильно диагностирует ошибку. Восстановление вернет квантовое состояние в кодовое подпространство, но в *закодированной* в блоке квантовой информации произойдет инвертирование бита:

$$|0\rangle_{\text{code}} \to |1\rangle_{\text{code}}, \quad |1\rangle_{\text{code}} \to |0\rangle_{\text{code}}. \tag{12}$$

Аналогично, если в одном блоке возникают две фазовые ошибки, то есть две ошибки инвертирования бита в повернутом базисе, то после восстановления в нем произойдет инвертирование бита в повернутом базисе или обращение фазы в исходном базисе:

$$|0\rangle_{\text{code}} \to |0\rangle_{\text{code}}, \quad |1\rangle_{\text{code}} \to -|1\rangle_{\text{code}}. \tag{13}$$

(Если один кубит в блоке содержит фазовую ошибку, а другой — ошибку инвертирования бита, то восстановление будет успешным.)

Итак, мы увидели, что код Стина способен повысить надежность хранения квантовой информации. Предположим, мы хотим сохранить один кубит в неизвестном чистом состоянии $|\psi\rangle$. Вследствие несовершенства запоминающего устройства состояние ρ_{out}, которое мы восстановим, будет иметь точность воспроизведения

$$F \equiv \langle\psi|\rho_{\text{out}}|\psi\rangle = 1 - \epsilon. \tag{14}$$

Но если мы храним кубит, используя 7-кубитовый блоковый код Стина, если каждый из семи кубитов хранится с точностью воспроизведения $F = 1-\epsilon$, если ошибки кубитов некоррелированы и, наконец, если мы в состоянии безукоризненно выполнять коррекцию ошибок, кодирование и декодирование (более подробно об этом ниже), то закодированная информация может храниться с улучшенной точностью воспроизведения $F = 1 - O(\epsilon^2)$.

Кубит в неизвестном состоянии можно закодировать, используя изображенную на рисунке 8 схему. Проще всего понять принцип работы кодирующего устройства, используя альтернативное выражение для матрицы контроля четности Хэмминга,

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}. \tag{15}$$

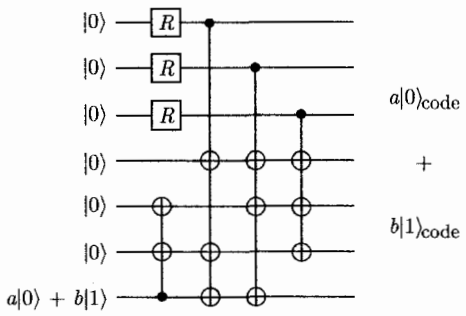


Рис. 8. Кодирующая схема для 7-кубитового кода Стина

(Эта форма матрицы H получена из (1) путем перестановки столбцов, то есть лишь изменением нумерации битов в блоке.) Четный субкод кода Хэмминга фактически является пространством, натянутым на строки матрицы H; итак, мы видим, что (в этом представлении матрицы H) первые три бита строки полностью характеризуют представленные в субкоде данные. Остальные четыре бита представляют собой биты четности, обеспечивающие необходимую для защиты от ошибок избыточность. При кодировании неизвестного состояния $a|0\rangle + b|1\rangle$ кодирующее устройство сначала использует два XOR-вентеля, чтобы приготовить состояние $a|0000000\rangle + b|0000111\rangle$, суперпозицию четного и нечетного слов Хэмминга. Далее по схеме к этому состоянию добавляется $|0\rangle_{\text{code}}$: повороты Адамара (R) готовят равновзвешенную суперпозицию всех восьми возможных значений первых трех битов в блоке, а остальные XOR-вентили включают предписываемые матрицей H биты контроля четности.

Мы также хотим иметь возможность измерять закодированный кубит, скажем, проецируя его на ортогональный базис $\{|0\rangle_{\text{code}}, |1\rangle_{\text{code}}\}$. Если нас не беспокоит разрушение закодированного блока в ходе измерения, то достаточно измерить каждый из семи кубитов в блоке, проецируя их на базис $\{|0\rangle, |1\rangle\}$, а затем, чтобы получить кодовое слово Хэмминга, выполнить классическую коррекцию ошибок в результатах измерения. Четность этого кодового слова является значением логического кубита. (Этап классической коррекции обеспечивает защиту от ошибок измерения. Например, если блок находится в состоянии $|0\rangle_{\text{code}}$, то при измерении элементарных кубитов для определения логического кубита могут возникнуть две независимые ошибки, в результате чего будет получено неверное значение $|1\rangle_{\text{code}}$.)

В применении к квантовым вычислениям, нам понадобится выполнять измерение, проецирующее на $\{|0\rangle_{\text{code}}, |1\rangle_{\text{code}}\}$, без разрушения блока. Это

выполняется путем копирования четности блока на служебный кубит с последующим его измерением. Схема, представляющая неразрушающее измерение кодового блока, показана на рисунке 9.

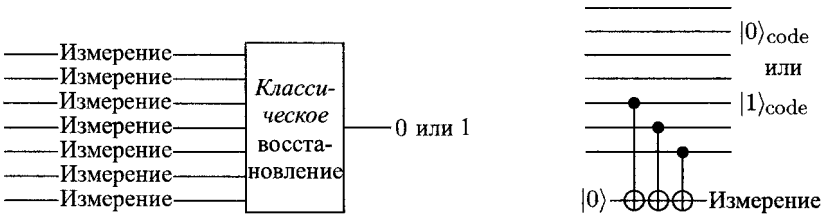


Рис. 9. Разрушающее и неразрушающее измерение логического кубита

7-кубитовый код Стина может может исправить только одну ошибку в кодовом блоке, но можно сконструировать более совершенные коды, способные защищать информацию от ошибок количеством до t внутри одного блока, так что закодированная информация может сохраняться с точностью воспроизведения $F = 1 - O(\epsilon^{t+1})$ (Steane 1996b; Calderbank & Shor 1996; Gottesman 1996; Calderbank *et al.* 1996, 1997). Обзор текущего состояния теории квантового кодирования сделан Шором в работе (Shor 1997).

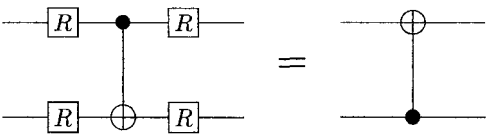


Рис. 10. Полезное тождество. Источник и цель XOR-вентиля меняются местами при изменении базиса с помощью поворотов Адамара

4. Отказоустойчивое восстановление

Конечно, исправление ошибок никогда не будет безупречным. Восстановление само по себе является предрасположенным к появлению ошибок квантовым вычислением. Необходимо позаботиться о создании процедуры восстановления для кода Стина, обеспечивающей вероятность возникновения ошибки порядка ϵ^2 (или порядка ϵ^{t+1} для кода, исправляющего t ошибок). Не менее серьезную проблему представляет контроль распространения ошибок в ходе процедуры восстановления.

Изображенная на рисунке 7 схема не является отказоустойчивой; она нарушает первый закон. XOR-вентили могут перенести единственную

ошибку, возникшую в одном из служебных кубитов, в два разных кубита в блоке данных, приводя в итоге к фазовой ошибке, появляющейся в закодированных данных с вероятностью порядка ϵ. Выполняя требования первого закона, следует увеличить количество служебных кубитов, так чтобы каждый из них являлся целью лишь одного XOR-вентиля. Но пока обратим внимание на второй закон: служебные кубиты должны запутываться с *ошибками* блока данных, а не с закодированной в нем квантовой информацией. Действительно, запутывание служебного кубита с закодированными данными разрушает их когерентность.

Выполняя это условие, мы, прежде чем приступать к процедуре вычисления синдрома, готовим *служебное состояние Шора* (Shor 1996). Это состояние четырех служебных кубитов, представляющее собой равновзвешенную суперпозицию всех строк с четным весом:

$$|\mathrm{Shor}\rangle_{\mathrm{anc}} = \frac{1}{\sqrt{8}} \sum_{v_{\mathrm{even}}} |v_{\mathrm{even}}\rangle_{\mathrm{anc}}. \qquad (16)$$

Чтобы вычислить каждый бит синдрома, мы готовим состояние Шора, выполняем четыре XOR-вентиля (с соответствующими кубитами в блоке данных в роли источников и с четырьмя служебными кубитами в состоянии Шора в роли целей), а затем измеряем служебное состояние. Бит синдрома извлекается из четности результата измерения состояния четырех служебных кубитов. Состояние Шора устроено таким образом, что из него можно извлечь *только* эту четность — никакой другой информации о блоке данных на нем не отпечатывается.

Всего имеется шесть битов синдрома (три для диагностики ошибок инвертирования бита и три для диагностики ошибок обращения фазы), так что измерение синдрома использует 24 служебных кубита, приготовленных в шести состояниях Шора, и 24 XOR-вентиля.

[Одним из способов получения синдрома обращения фазы может быть следующий: сначала применить семь параллельных R-вентилей к блоку данных, чтобы повернуть базис, затем, как показано на рисунке 7, применить XOR-вентили (но с приготовленными в состояниях Шора служебными кубитами) и, наконец, применить семь R-вентилей для обратного поворота данных. Но чтобы усовершенствовать эту процедуру, мы можем использовать представленное на рисунке 10 тождество. Обращая направление XOR-вентилей (то есть используя служебные кубиты в качестве источников, а информацию — в качестве цели), мы можем избежать применения R-вентилей к данным и, следовательно, понизить вероятность их повреждения неправильными вентилями (Zalka 1996; Steane 1997).]

Вследствие распространения ошибок, единственная возникающая в процессе приготовления состояния Шора ошибка может стать результатом возникновения двух фазовых ошибок в данном состоянии. Обе они могут распространиться на данные, если при измерении синдрома используется поврежденное служебное состояние. Следовательно, перед использованием состояния Шора оно должно быть протестировано на наличие нескольких фазовых ошибок (пример третьего закона), как это показано на рисунке 11. Если состояние не проходит тест, его следует уничтожить, и приготовить новое.

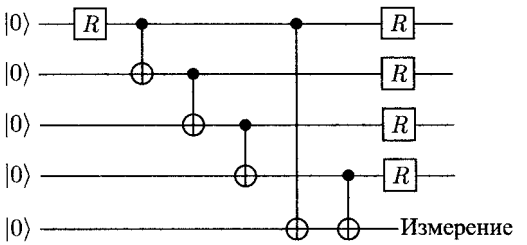


Рис. 11. Конструирование и проверка состояния Шора. Если причиной измерения является 1, то состояние отбрасывается, и готовится новое состояние Шора

Наконец, единственная ошибка в процессе измерения синдрома может стать причиной появления ошибочного синдрома. Таким образом, для подтверждения точности измерение синдрома необходимо повторить (четвертый закон). Когда один и тот же результат получается дважды подряд, он может быть принят с уверенностью, а восстановление продолжено.

Если мы применяем все описанные выше меры предосторожности, то восстановление дает сбой только при появлении двух независимых ошибок, поэтому вероятность возникающей в закодированном блоке ошибки будет порядка ϵ^2. Процедура коррекции ошибок является отказоустойчивой.

Довольно изящная отказоустойчивая процедура измерения синдрома была предложена Стином (Steane 1997). Для измерения синдрома инвертирования бита готовится служебное 7-кубитовое *состояние Стина*

$$|\text{Steane}\rangle_{\text{anc}} = \frac{1}{4}\sum_{v\in H}|v\rangle. \tag{17}$$

(Состояние Стина можно приготовить, применив к состоянию $|0\rangle_{\text{code}}$ побитовый поворот Адамара.) Теперь выполним операции XOR с каждым кубитом блока данных в качестве источника и соответствующим кубитом

служебного состояния в качестве цели, после чего измерим результат действия этих операций на данное служебное состояние. Применяя матрицу контроля четности Хэмминга H к результату *классического* измерения, мы получаем синдром инвертирования бита. (Обратите внимание на соблюдение второго закона: процедура «копирует» данные в служебный блок, состояние которого устроено таким образом, что при его измерении обеспечивается возможность прочтения лишь информации об ошибке.) Аналогичная процедура выполняется в повернутом базисе для определения синдрома обращения фазы. Преимущество процедуры Стина перед процедурой Шора состоит в том, что она требует лишь 14 служебных кубитов и 14 XOR-вентилей. Однако недостаток ее заключается в том, что приготовление служебного состояния более сложное, так что оно само в некоторой степени больше предрасположено к возникновению ошибок.

А что можно сказать об измерении и кодировании? Мы только что отметили, что разрушающее измерение кодового блока надежно, если ошибка содержится только в одном кубите блока. Изображенное на рисунке 9 не разрушающее информацию измерение также не требует модификации. Несмотря на то, что служебное состояние является целью трех последовательных XOR-вентилей, возвращающиеся в блок фазовые ошибки не опасны, так как они не могут поменять $|0\rangle_{\text{code}}$ на $|1\rangle_{\text{code}}$ (или наоборот). Но, поскольку единственная ошибка может оказаться причиной неправильного результата измерения четности, его необходимо повторить (после исправления ошибок), чтобы обеспечить точность порядка ϵ^2 (пример четвертого закона).

В применении к квантовым вычислениям нам потребуется повторно готовить закодированное состояние $|0\rangle_{\text{code}}$. Кодирование может быть выполнено по изображенной на рисунке 8 схеме (за исключением того, что первые два XOR-вентиля можно исключить). Однако ошибки могут распространяться в ходе процедуры кодирования, так что единственной ошибки может быть достаточно для повреждения закодированной информации. Следовательно, важно проверять кодирование, как этого требует третий закон, выполняя неразрушающее измерение блока. На самом деле можно вполне обойтись без этапа кодирования. Каким бы ни было исходное состояние блока, отказоустойчивая коррекция ошибок спроецирует его на пространство, натянутое на $\{|0\rangle_{\text{code}}, |1\rangle_{\text{code}}\}$, а (проверенное) измерение выдаст либо $|0\rangle_{\text{code}}$, либо $|1\rangle_{\text{code}}$. Если получен результат $|1\rangle_{\text{code}}$, то для перевода блока в желаемое состояние $|0\rangle_{\text{code}}$ можно применить (побитовый) оператор NOT.

Кодирование неизвестного квантового состояния выполняется с помощью изображенной на рисунке 8 схемы. Вновь, вследствие распростране-

ния ошибок, единственная ошибка, возникшая в ходе кодирования, может оказаться причиной его сбоя. В этом случае, поскольку никаким измерением нельзя проверить результат кодирования, точность воспроизведения закодированного состояния будет равна $F = 1 - O(\epsilon)$. Тем не менее, смысл в кодировании остается, поскольку закодированное состояние может храниться с требуемой надежностью в течение более продолжительного времени, чем незакодированное.

Обе схемы (Шора и Стина) отказоустойчивого измерения синдрома описаны здесь только для 7-кубитового кода, но их можно приспособить для более сложных кодов, способных корректировать множество ошибок (DiVincenzo & Shor 1996; Steane 1997). По мере возрастания сложности кода схема Стина становится существенно эффективнее схемы Шора.

5. Отказоустойчивые квантовые вентили

Мы убедились, что кодирование может защитить квантовую информацию. Но мы хотим больше, нежели просто *хранить* квантовую информацию с высокой точностью воспроизведения; мы хотим управлять квантовым компьютером, который *обрабатывает* информацию. Конечно, мы могли бы декодировать информацию, выполнить вентиль, а затем кодировать ее заново, но эта процедура на время подвергла бы опасности информацию. Напротив, если мы хотим, чтобы наш компьютер работал надежно, мы должны уметь применять квантовые вентили непосредственно к закодированной информации, а эти вентили должны удовлетворять первому закону отказоустойчивости, если мы хотим избежать катастрофического распространения ошибок.

Действительно, для 7-кубитового кода Стина существует несколько вентилей, которые можно легко применить. Три однокубитовых вентиля могут применяться *побитово*, то есть применение этих вентилей к каждому из семи кубитов в блоке осуществляет тот же вентиль, действующий на закодированный кубит. Мы уже увидели в уравнении (11), что именно так действует поворот Адамара R. То же самое справедливо для NOT-вентиля (поскольку каждое нечетное кодовое слово Хэмминга является дополнением четного кодового слова Хэмминга),[1] а также и вентиля сдвига фазы

$$P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}; \tag{18}$$

нечетные кодовые слова Хэмминга имеют вес $\equiv 3$ (по модулю 4), а четные кодовые слова имеют вес $\equiv 0$ (по модулю 4), поэтому мы фактически при-

[1] Собственно, мы можем выполнить операцию NOT, действующую на закодированный кубит, при помощи лишь трех NOT, применяемых к выбранным кубитам в блоке.

меняем вентиль P^{-1} побитово, чтобы выполнить вентиль P. XOR-вентиль также может выполняться побитово, то есть используя каждый бит блока источника и соответствующий бит блока цели. Это работает, потому что четные кодовые слова образуют субкод, тогда как нечетные кодовые слова представляют его нетривиальный смежный класс.

Таким образом, существуют простые отказоустойчивые процедуры для выполнения NOT-, R-, P- и XOR-вентилей. Но, к сожалению, сами по себе эти вентили не образуют универсального набора. Чтобы уметь выполнять произвольные унитарные преобразования закодированной квантовой информации, нам потребуется сделать подходящее дополнение этого набора. Следуя Шору (Shor 1996), мы добавим 3-кубитовый вентиль Тоффоли, который выполняется с помощью процедуры, изображенной на рисунке 12.[1]

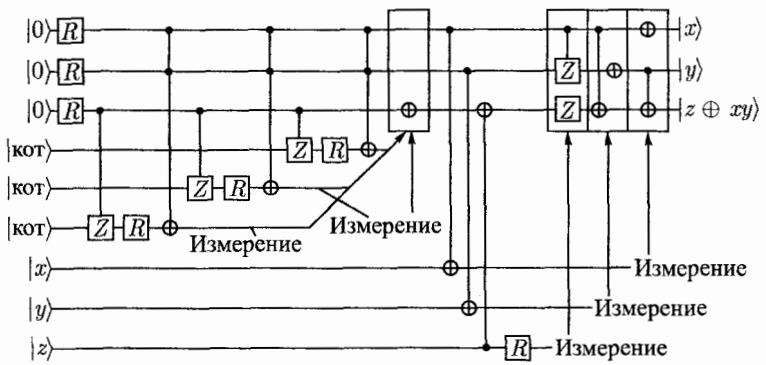


Рис. 12. Отказоустойчивый вентиль Тоффоли. Каждая линия изображает блок из семи кубитов, а вентили применяются трансверсально. Для каждого измерения стрелка указывает на набор вентилей, которые применяются, если результатом измерения является 1, и не действуют, если результатом является 0

Вкратце, процедура работает следующим образом. Во-первых, три закодированных служебных блока приготавливаются в состоянии вида

$$|A\rangle_{\text{anc}} \equiv \sum_{a=0,1} \sum_{b=0,1} |a, b, ab\rangle_{\text{anc}}. \tag{19}$$

[1] Нилл и другие (Knill *et al.* 1996, 1997) описывает альтернативный способ дополнения универсального набора вентилей.

Это приготовление служебного блока выполняется с помощью (проверенного) 7-битового «кот-состояния»

$$|\text{cat}\rangle = \frac{1}{\sqrt{2}}\big(|0000000\rangle + |1111111\rangle\big). \tag{20}$$

Выполняется несколько вентилей, включая побитовый вентиль Тоффоли с двумя служебными блоками в качестве управляющих и кот-состоянием в качестве цели. Затем измеряется кот-состояние. (Измерение повторяется, чтобы удостовериться в его точности.) Если результаты измерений четные, значит, желаемое вспомогательное состояние $|A\rangle_{\text{anc}}$ успешно приготовлено; если нечетные, то состояние $|A\rangle_{\text{anc}}$ можно получить, применив NOT-вентиль к третьему служебному блоку.

Тем временем три блока данных терпеливо ожидали готовности служебного блока. Применением трех XOR-вентилей и поворота Адамара состояние данных и служебного блока преобразуется следующим образом:

$$\begin{aligned}&\sum_{a=0,1}\sum_{b=0,1}|a,b,ab\rangle_{\text{anc}}|x,y,z\rangle_{\text{data}} \rightarrow \\ &\quad\rightarrow \sum_{a=0,1}\sum_{b=0,1}\sum_{w=0,1}(-1)^{wz}|a,b,ab\oplus z\rangle_{\text{anc}}|x\oplus a, y\oplus b, w\rangle_{\text{data}}.\end{aligned} \tag{21}$$

Теперь выполняется измерение каждого блока *данных*. Если результатом измерения является 0, никакие действия не предпринимаются, но если результат измерения — 1, то для завершения выполнения вентиля Тоффоли к *служебному* блоку применяется указанный на рисунке 12 набор вентилей. Обратите внимание на то, что исходные блоки данных разрушаются данной процедурой, а также на то, что новыми блоками данных становятся блоки, первоначально бывшие служебными. Важным свойством этой конструкции является то, что все ее этапы организованы так, чтобы удовлетворять требованиям законов отказоустойчивости.

Немного мешает то, что отказоустойчивые вентили формируют дискретный набор, но в то же время это неизбежная черта любой отказоустойчивой схемы. Для отказоустойчивых вентилей нет смысла формировать континуум, ибо как тогда мы сможем избежать совершения ошибки, применяя *ложный* вентиль, который отличается от предназначенного на небольшую величину? В любом случае, поскольку наши отказоустойчивые вентили формируют универсальный набор, их достаточно для приближенного выполнения любого желаемого унитарного преобразования с любой желаемой точностью.

Шор показал, как обобщить этот отказоустойчивый набор вентилей на более сложные коды, способные исправлять большее количество ошибок, а Готтесман (Gottesman 1997ab) описал еще более общую процедуру, которую можно применять к любому из известных квантовых кодов. Таким образом, практически любой корректирующий ошибки квантовый код может использоваться для отказоустойчивых вычислений. В чем же тогда смысл пятого закона? Даже если, в принципе, может использоваться любой код, некоторые из них работают эффективнее. Например, существует 5-кубитовый код, способный корректировать одну ошибку (Bennett *et al.* 1996; Laflamme *et al.* 1996), а Готтесман представил универсальный набор отказоустойчивых вентилей для этого кода. Но реализация этих вентилей достаточно сложна. Для 7-кубитового кода Стина требуется больший блок, но он гораздо удобнее для реализации вычислений; пятый закон отдает предпочтение коду Стина.

6. Порог безошибочности квантовых вычислений

Существуют корректирующие квантовые коды, способные исправить t ошибок, где t может быть сколь угодно большим. Если мы используем такой код и следуем законам отказоустойчивости, тогда непоправимая ошибка возникнет только при появлении $t+1$ *независимых* ошибок в одном блоке до завершения восстановления. Поэтому если вероятность возникновения ошибки в одном квантовом вентиле или отнесенная к единице времени вероятность возникновения ошибки запоминающего устройства имеет порядок ϵ, то вероятность действующей на закодированные данные ошибки в вентиле будет иметь порядок ϵ^{t+1}, что гораздо меньше, чем ϵ, если эта величина достаточно мала. Действительно, может показаться, что при выборе кода со сколь угодно большим t мы можем сделать вероятность ошибки в вентиле сколь угодно малой, но это не факт, по крайней мере для большинства кодов. Проблема состоит в том, что по мере увеличения t сложность кода резко возрастает, соответственно возрастает и сложность процедуры восстановления. В конечном счете мы достигаем момента, когда выполнение восстановления требует так много времени, что становится вероятным накопление $t+1$ ошибок в блоке до завершения восстановительного этапа, и, таким образом, способность кода исправлять ошибки становится сомнительной.

Предположим, что количество вычислительных шагов, необходимых для выполнения измерения синдрома, растет вместе с t, как степень t^b. Тогда вероятность того, что до завершения измерения накопится $t+1$ ошибок, будет вести себя как

$$\text{Block Error Probability} \sim \left(t^b\epsilon\right)^{t+1}, \tag{22}$$

где ϵ — вероятность ошибки на один шаг. Мы можем в таком случае выбрать t, чтобы минимизировать вероятность ошибки ($t \sim e^{-1}\epsilon^{-1/b}$, при условии, что t большое), получая

$$\text{Minimum Block Error Probability} \sim \exp\left(-e^{-1}b\epsilon^{-1/b}\right). \tag{23}$$

Таким образом, если мы рассчитываем безупречно выполнить в совокупности T циклов коррекции ошибок, то наши вентили должны иметь точность

$$\epsilon \sim (\log T)^{-b}. \tag{24}$$

Аналогично, для выполнения квантового вычисления с участием T квантовых вентилей необходимы элементарные вентили заданной точности.

В первоначально описанной Шором (Shor 1996) процедуре степень, характеризующая сложность измерения синдрома, равна $b = 4$; с помощью более оптимизированной процедуры можно достичь скромного улучшения ($b \sim 3$). Размер блока используемого кода растет вместе с t как t^2, поэтому, когда выбирается код для оптимизации вероятности возникновения ошибки, размер блока будет порядка $(\log T)^2$. Конечно, описываемый уравнением (24) скейлинг гораздо предпочтительнее, чем точность $\epsilon \sim T^{-1}$, которая потребовалась бы, если бы кодирование не использовалось вообще. Но при любой заданной точности существует предел продолжительности вычисления, при которой еще можно не опасаться появления ошибок.

Это ограничение можно преодолеть, используя код специального типа, а именно *каскадный* код (Knill & Laflamme 1996; Knill *et al.* 1996, 1997; Aharonov & Ben-Or 1996; Kitaev 1996, 1997). Чтобы понять идею каскадного кода, представьте, что мы используем корректирующий ошибки квантовый код Стина, кодирующий единственный кубит в 7-кубитовом блоке. Но если мы посмотрим внимательнее, с бóльшим разрешением, на один из семи кубитов в блоке, то обнаружим, что на самом деле это не один кубит, а другой 7-кубитовый блок, закодированный, как и ранее, при помощи того же кода Стина. А когда мы изучим один из семи кубитов в *этом* блоке с еще бóльшим разрешением, мы обнаружим, что он тоже в действительности является блоком из семи кубитов, и так далее (см. рис. 13). Если в этой иерархии каскадного соединения всего имеется L уровней, тогда один кубит фактически кодируется в блоке размера 7^L. Каскадное соединение оказывается полезным, поскольку, действуя по принципу «разделяй и властвуй», оно позволяет более эффективно избавляться от ошибок: то есть чаще всего восстановление осуществляется на самом нижнем уровне иерархии, в блоке размера семь, реже — на следующем уровне, в блоке размера $7^2 = 49$, еще реже на следующем уровне, на блоке размера $7^3 = 343$,

и так далее. При таком методе сложность коррекции уже не так стремительно растет с повышением способности квантового кода исправлять ошибки.

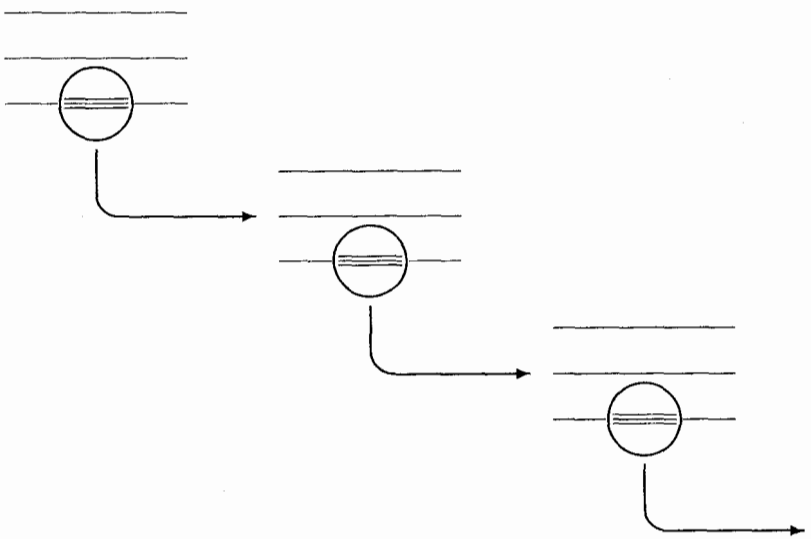


Рис. 13. Каскадное кодирование. Каждый кубит в блоке, рассматриваемый с большим разрешением, сам является закодированным субкодом

Мы видели, что 7-кубитовый код Стина может исправлять одну ошибку. Если вероятность ошибки на кубит равна ϵ, ошибки не коррелированы, а процедура отказоустойчива, то вероятность сбоя при восстановлении будет порядка ϵ^2. Если два кода соединяются в каскад, образуя блок размера 7^2, то ошибка в нем возникает только при повреждении двух из его субблоков размера семь, что происходит с вероятностью порядка $(\epsilon^2)^2$. При добавлении еще одного уровня каскадирования ошибка в блоке размера 7^3 возникает только в случае повреждения двух из его субблоков размера 7^2, а вероятность такого события имеет порядок $\left((\epsilon^2)^2\right)^2$. И так далее, при наличии L уровней каскадного соединения кодов вероятность появления ошибки имеет порядок ϵ^{2^L}, тогда как размер блока равен 7^L. Теперь, если частота появления ошибок для основных вентилей достаточно мала, то отнесенную к одному вентилю вероятность появления ошибки можно уменьшить с помощью каскадного соединения кодов. Если это так, то добавление

следующего уровня каскадирования приведет к дальнейшему понижению вероятности ошибки, и так далее. В этом состоит природа порога безошибочности квантовых вычислений: если кодирование значительно снижает вероятность возникновения ошибки, то, добавляя достаточное количество уровней каскадирования, частоту появления ошибок можно сделать сколь угодно малой. Но если исходная частота появления ошибок слишком высока, то кодирование, напротив, усугубит такое положение вещей.

Чтобы проанализировать эту ситуацию, необходимо принять некоторую конкретную модель ошибок. Я выберу наиболее простую из возможных квазиреалистичную модель: некоррелированные стохастические ошибки.[1] На каждом такте вычисления каждый кубит в устройстве запутывается с окружением. Пусть запутывание описывается уравнением (4) с тем лишь отличием, что теперь четыре состояния окружения предполагаются взаимно ортогональными, а «ошибочные состояния» — имеющими одинаковые нормы. Таким образом, три типа ошибок (инвертирование бита, обращение фазы и обе ошибки одновременно) предполагаются равновероятными. Полная вероятность появления ошибки на каждом шаге обозначается как ϵ_{store}. Кроме этих ошибок запоминающего устройства, поражающих хранящиеся кубиты, существуют ошибки, вносимые самими квантовыми вентилями. Для каждого типа вентиля вероятность появления ошибки при каждом его применении обозначается как ϵ_{gate} (при независимых значениях, приписываемых каждому типу вентилей). Если вентиль действует на более чем один кубит (XOR или Тоффоли), то могут возникать коррелирующие ошибки. Сделаем пессимистическое предположение, что ошибка многокубитового вентиля всегда повреждает все кубиты, на которые он действует; например, неправильный XOR-вентиль вводит ошибки одновременно в кубит источника и кубит цели. Это допущение (среди прочих) делается только для упрощения анализа. При более реалистичных предположениях мы, безусловно, обнаружили бы, что вполне можно пережить и несколько более высокую частоту появления ошибок.

Эффективность каскадного кодирования можно проанализировать, построив систему *потоковых уравнений*, описывающих эволюцию модели ошибки при переходе от одного уровня каскадирования к другому. Пусть, например, мы хотим выполнить XOR-вентиль с последующим этапом исправления ошибок в кубитах, закодированных с помощью каскадного кода Стина, содержащего L уровней (размер блока 7^L). Эти процедуры можно описать с помощью таких же операций, но действующих на субблоки размера 7^{L-1}. Таким образом, вероятность появления ошибки $\epsilon^{(L)}$ для вен-

[1] В разделе 9 я прокомментирую, как изменяется анализ при выборе другой модели ошибок.

тиля, действующего на блок размера L, может быть выражена через вероятность появления ошибки $\epsilon^{(L-1)}$ для вентиля, действующего на блок размера $L-1$. Это соотношение представляет собой одно из потоковых уравнений. В принципе, решая систему потоковых уравнений, можно получить выражение для вероятности появления ошибки «на уровне L» через параметры модели ошибок и исследовать ее поведение с ростом L. Если вероятность появления ошибки в блоке стремится к нулю с ростом его размера L, то это это означает, что вероятности элементарных ошибок лежат «ниже порога». Поскольку вероятности элементарных ошибок могут зависеть от множества параметров, то на самом деле порог представляет собой некоторую гиперповерхность в многомерном параметрическом пространстве рассматриваемой модели.

В общем случае потоковые уравнения довольно сложны; их подробное обсуждение можно найти в работах Готтесмана (Gottesman 1997b), Готтесмана и других (Gottesman *et al.* 1996) или Залки (Zalka 1996). Но для грубой иллюстрации идеи предположим, что ошибки хранения отсутствуют, а единственным источником ошибок являются XOR-вентили, характеризуемые отнесенной к одному вентилю вероятностью появления ошибки $\epsilon_{\rm XOR}$. Допустим, мы хотим применить сеть XOR-вентилей, действующих на закодированные блоки. Оценим, насколько малой должна быть частота появления ошибок $\epsilon_{\rm XOR}$, достигаемая действующими на закодированную информацию вентилями, чтобы эта величина была меньше частоты ошибок для элементарных XOR-вентилей.

Если воспользоваться состоянием Шора в качестве служебного, то для выявления синдромов инвертирования бита и обращения фазы потребуется 12 XOR-вентилей,[1] следовательно, вероятность появления ошибки для каждого измерения синдрома равна

$$\epsilon_{\rm syndrome} \sim 12\epsilon_{\rm XOR}. \tag{25}$$

Чтобы убедиться в надежности синдрома, он повторно измеряется до тех пор, пока дважды подряд не получается один и тот же результат.[2] Из простой комбинаторики можно найти, что вероятность возникновения двух независимых ошибок до успешного завершения измерения синдрома равна $5\epsilon^2_{\rm syndrome}$. Предположим, что обе эти ошибки повреждают данные, тогда они являются причиной появления непоправимой ошибки в блоке. Поскольку нам нужно измерить как синдром инвертирования бита, так и син-

[1] При использовании служебного состояния Стина это число было бы сокращено до 7; однако, поскольку приготовление состояния Стина сложнее, служебное состояние будет в большей степени подвержено ошибкам.

[2] На самом деле это не наилучшая процедура, но здесь мы ее примем для простоты.

дром обращения фазы, мы видим, что вероятность сбоя при каждом выполнении исправления равна

$$\epsilon_{\text{fail}} \sim 2 \cdot 5 \cdot (12\epsilon_{\text{XOR}})^2 = 1440\epsilon_{\text{XOR}}^2. \tag{26}$$

При реализации сети XOR-вентилей мы выполняем исправление в каждом блоке после того, как на него подействовали N вентилей, где N — число, которое будет выбрано для оптимизации количества ошибок на вентиль. Каждый трансверсальный XOR-вентиль, действующий на блок, выполняется путем применения семи элементарных XOR-вентилей. После применения к данному блоку N вентилей и следующего вслед за этим этапа восстановления, отнесенная к одному вентилю вероятность накопления двух ошибок в блоке не превосходит

$$\epsilon_{\text{fail}} \sim \frac{1}{N}\left(\frac{1}{2}7N \cdot 6N \cdot \epsilon_{\text{XOR}}^2 + 7N \cdot \epsilon_{\text{XOR}} \cdot 2 \cdot 2 \cdot \epsilon_{\text{syndrome}} + 1440 \cdot \epsilon_{\text{XOR}}^2\right) =$$

$$= \frac{1}{N}\left(21N^2 + 336N + 1440\right)\epsilon_{\text{XOR}}^2. \tag{27}$$

(Возможны три варианта: либо обе ошибки внесены вентилями, либо одна из них происходит от вентиля, а другая возникла на этапе восстановления, либо наконец, обе они возникли в процессе исправления.[1] Мы пессимистично предполагаем, что все ошибки повреждают данные в блоке и что при возникновении двух ошибок в блоке данных исправление всегда дает сбой.) Минимум в уравнении (27) достигается при $N = 8$, тогда мы получаем $\epsilon_{\text{fail}} = 684\epsilon_{\text{XOR}}^2$, то есть кодирование имеет смысл при $684\epsilon_{\text{XOR}}^2 < \epsilon_{\text{XOR}}$, или $\epsilon_{\text{XOR}} < (684)^{-1} \sim 1.5 \cdot 10^{-3}$. Это наша «пороговая оценка».

Даже если бы все ошибки возникали из-за XOR-вентилей, по ряду причин эта оценка была бы неадекватна. Более того, мы предположили, что служебные состояния (состояния Шора), используемые при измерении синдрома, свободны от ошибок, в то время как для их приготовления и проверки фактически используются те же самые XOR-вентили. К тому же, когда XOR-вентили действуют на субблоки каскадного кода, не всегда возможно применение оптимального числа вентилей перед исправлением. И, конечно, более полный анализ должен включать ошибки запоминающего устройства и отслеживать эволюцию связанных потоков ошибок хранения и ошибок вентилей, обусловленную добавлением очередного уровня каскадирования.

[1] Из двух коэффициентов 2 во втором слагаемом уравнения (27) один возникает вследствие того, что измеряются оба синдрома — инвертирования бита и обращения фазы, второй — потому что ошибка может возникнуть как в ходе первого измерения синдрома, так и при его повторении.

Действительно, когда мы осуществляем исправление на L-уровне, состояние Шора должно быть построено из блоков, закодированных на $(L-1)$-уровне. Важной частью анализа является тщательный контроль того, насколько шумящими являются эти служебные состояния, с целью определить вероятность отказа при исправлении на L-уровне. Блоки данных должны ожидать приготовления служебных состояний; тем временем в них накапливаются ошибки хранения. А так как период ожидания не масштабируется просто уровнем L, потоковые уравнения не являются полностью автомодельными — существует некоторая явная «временная зависимость» потока (то есть L-зависимость). Но по-прежнему качественно верно, что порог определяется требованием, чтобы отказоустойчивая процедура с одним уровнем каскадирования действительно улучшала надежность вентилей, как в приведенном выше примере.

Хотя потоковые уравнения слишком сложны для точного решения, при умеренных предположениях можно получить их приближенные решения. Грубо говоря, выводы следующие (Gottesman *et al.* 1996; Gottesman 1997b). Если ошибки хранения пренебрежимо малы, то пороговая частота появления ошибок на один вентиль имеет порядок 10^{-4}. Если доминируют ошибки хранения, то пороговая частота появления ошибок в течение одного такта имеет порядок 10^{-5} (где один такт представляет собой интервал времени, необходимого для выполнения одного вентиля). От элементарных вентилей Тоффоли не требуется такая точность, как от одно- и двухкубитовых вентилей, — вполне приемлема частота ошибок вентиля Тоффоли порядка 10^{-3}, если достаточно малы другие частоты появления ошибок. (Это приятно, поскольку вентили Тоффоли наиболее сложны в применении, и на практике вероятна их меньшая точность.)

Также следует задать вопрос, какой размер блока необходим для обеспечения некоторой определенной точности. Иначе говоря, если пороговая частота ошибки в вентиле равна ϵ_0, а фактическая частота ошибки в элементарном вентиле $\epsilon < \epsilon_0$, то L-кратное каскадирование кода понизит частоту возникновения ошибок до

$$\epsilon^{(L)} \sim \epsilon_0 \left(\frac{\epsilon}{\epsilon_0}\right)^{2^L}. \tag{28}$$

Таким образом, чтобы быть достаточно уверенными в том, что мы можем завершить вычисление с T вентилями без единой ошибки, мы должны выбрать размер блока 7^L порядка

$$\begin{matrix}\text{block}\\ \text{size}\end{matrix} \sim \left[\frac{\log \epsilon_0 T}{\log \epsilon_0/\epsilon}\right]^{\log_2 7}. \tag{29}$$

Если каскадный код имеет размер блока n и может исправить $t+1$ ошибок, показатель степени $\log_2 7 \sim 2.8$ в уравнении (29) заменяется на $\log n / \log(t+1)$; для семейства кодов, рассмотренных Шором, этот показатель стремится к 2, но для «хороших» кодов, в принципе, может стремиться к 1.

Если частоты ошибок лежат ниже порога безошибочности, также возможно сколь угодно продолжительное хранение *неизвестного* квантового состояния. Однако, как мы уже отмечали в разделе 4, если вероятность ошибки запоминающего устройства на один такт вычисления равна ϵ, то исходное кодирование состояния может быть выполнено с точностью воспроизведения не выше, чем $F = 1 - O(\epsilon)$. При каскадном кодировании мы можем бесконечно долго хранить неизвестную квантовую информацию с достаточно хорошей, но не со сколь угодно высокой, точностью воспроизведения.

Допущения, лежащие в основе этих выводов, будут еще раз рассмотрены в разделе 9.

7. Отказоустойчивая факторизация

Чтобы понять практическое значение скейлингого закона (29), рассмотрим применение каскадного кодирования для реализации квантового алгоритма факторизации Шора (Shor 1994). Алгоритм состоит из двух частей. Для факторизации числа N сначала вычисляется показательная функция по модулю N, чтобы приготовить состояние вида

$$\sum_x |x\rangle_{\text{input}} \otimes |a^x \pmod N\rangle_{\text{output}}, \tag{30}$$

где $a\,(< N)$ — взаимно простое с N. Затем выполняется преобразование Фурье, действующее на входной регистр. Наконец, входной регистр измеряется и выполняется некоторая классическая постобработка, чтобы найти кандидата на роль простого множителя числа N. Приготовление состояния (30) (с применением перечисленных в разделе 5 отказоустойчивых вентилей) описано в работах Бэкмана и др. (Beckman *et al.* 1996) и Ведрала и др. (Vedral *et al.* 1996). Преобразование Фурье требует комментариев. Как показано Гриффитсом и Ниу (Griffits & Niu 1996), преобразование Фурье может быть измерено с помощью однокубитовых вентилей (но при условии, что тип вентиля определяется результатами предыдущих измерений). В частности, для этого необходимо уметь выполнять вентили поворота фазы

$$P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}. \tag{31}$$

Впрочем, достаточно иметь в распоряжении отказоустойчивую процедуру реализации $P(\theta_0)$ при некотором значении θ_0, иррациональном кратном 2π. Тогда повторное применение $P(\theta_0)$ позволяет сколь угодно точно выполнить $P(\theta)$ при любом θ.

Чтобы построить $P(\theta_0)$, необходимо использовать два служебных кубита и применить два вентиля Тоффоли. Одна из рабочих схем изображена на рисунке 14.[1] Если результатом измерения двух служебных кубитов является $|00\rangle_{\text{ancilla}}$, что происходит с вероятностью $5/8$, то это означает, что схема успешно применила $P(\theta_0)$ к информационному кубиту, где $e^{i\theta_0} = (1+3i)/(3+i)$, или $\cos\theta_0 = 3/5$. Любой другой результат ($|01\rangle$, $|10\rangle$ или $|11\rangle$, каждый из которых возникает с вероятностью $1/8$), означает сбой в работе схемы. Но в этом случае, применяя Z-вентиль, можно восстановить состояние поврежденного кубита и продолжить попытки применения этой схемы вплоть до их успешного завершения.

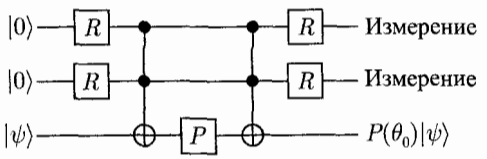


Рис. 14. Схема фазовращателя. Результат измерения $|00\rangle$ означает, что к кубиту данных применен поворот $P(\theta_0)$, где $\cos\theta_0 = 3/5$

В качестве альтернативного метода можно собрать «автономную» библиотеку «угловых кубитов» вида

$$|\theta\rangle = \frac{1}{\sqrt{2}}\bigl(|0\rangle + e^{i\theta}|1\rangle\bigr). \tag{32}$$

Тогда фазовращатель $P(\theta)$ можно реализовать, применяя вентиль CNOT с кубитом данных в качестве управляющего и библиотечным угловым кубитом $|\theta\rangle$ в качестве цели, как это показано на рисунке 15. Затем мы измеряем библиотечный кубит. Результат измерения $|0\rangle$ получается с вероятностью $\frac{1}{2}$, и в этом случае мы успешно применили $P(\theta)$ к информационному кубиту. Но если в результате измерения получено $|1\rangle$, то это означает, что вместо $P(\theta)$ к состоянию $|\psi\rangle$ применено преобразование $P(-\theta)$. В этом случае

[1] Кроме двух вентилей Тоффоли, на этой схеме изображены четыре вентиля Адамара, обозначаемые здесь как R, и фазовращатель $P\,[= P(\pi/4)]$. — *Прим. ред.*

предпринимается еще одна попытка, но на этот раз, чтобы компенсировать ошибку предыдущего шага, применяется схема $P(2\theta)$. Вероятность n сбоев подряд равна всего лишь 2^{-n}.

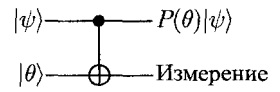


Рис. 15. Схема фазовращателя, использующая в качестве служебного библиотечный кубит. Если результатом измерения является $|0\rangle$, то $P(\theta)$ успешно применен к кубиту данных

Теперь попытаемся факторизовать K-битовое число, используя схему факторизации Бэкмана и др. (Beckman *et al.* 1996) и первый из описанных выше методов выполнения однокубитовых поворотов, требуемых для преобразования Фурье. Чтобы выполнить преобразование Фурье (на регистре с $2K$ кубитами), необходимы $2K$ однокубитовых фазовращателей. Преобразование Фурье должно вычисляться с высокой точностью. Для того чтобы алгоритм факторизации имел достаточные шансы на успех, достаточно выполнение каждого фазового поворота с точностью порядка K^{-1}. Мы можем достичь этой точности, компонуя $P(\theta_0)$ порядка K раз. Поскольку преобразование Фурье требует порядка K^2 вентилей Тоффоли, то для большого K сложность данного алгоритма обусловлена вычислением модулярной показательной функции, которая, согласно Бэкману и др. (Beckman *et al.* 1996)[1], требует $38K^3$ вентилей Тоффоли.

При наличии лучших из известных классических алгоритмов и самых быстродействующих из существующих машин факторизация 130-разрядного числа ($K \sim 430$ бит) займет порядка нескольких месяцев (Lenstra *et al.* 1996). Зададим вопрос, какими возможностями должен обладать квантовый компьютер для выполнения этого задания. От него потребуется способность хранить $5K \sim 2150$ закодированных кубитов, а также выполнить порядка $3 \cdot 10^9$ вентилей Тоффоли. Чтобы достичь достаточно высокой вероятности выполнения безошибочного вычисления, нам бы хотелось, чтобы вероятность появления ошибки на один вентиль Тоффоли была менее 10^{-9}, а вероятность ошибки хранения на полное время выполнения вентиля — менее 10^{-12}. Согласно потоковым уравнениям каскадирования, такие частоты возникновения ошибок могут быть достигнуты для закодиро-

[1]Описанный Бэкманом и др. алгоритм фактически требует $46K^3$ вентилей Тоффоли, но это число может быть снижено до $38K^3$ при использовании предложенного Ричардом Хагесом (Hughes 1997) усовершенствованного алгоритма сравнения.

ванных данных, если частоты ошибок на уровне индивидуальных кубитов равны $\epsilon_{\text{store}} \sim \epsilon_{\text{gate}} \sim 10^{-6}$ и если используются три уровня каскадирования, так что размер кодирующего каждый кубит блока равен $7^3 = 343$. С учетом дополнительных служебных кубитов, необходимых для выполнения вентилей и (параллельного) исправления ошибок, общее число содержащихся в машине кубитов будет порядка 10^6.

При частоте ошибок порядка 10^{-6} для индивидуальных кубитов, каскадное соединение 7-кубитового кода может оказаться наиболее эффективной отказоустойчивой процедурой. Для частот появления ошибок меньшего порядка величины лучше воспользоваться более сложным (некаскадным) кодом, способным исправить несколько ошибок в одном блоке (Shor 1996). При еще более низких частотах появления ошибок можно использовать коды, которые более эффективно используют пространство памяти путем кодирования множества кубитов в одном блоке (Gottesman 1997a). В принципе, когда частота ошибок для индивидуальных кубитов очень низкая для выполнения отказоустойчивого вычисления становится возможным найти хороший код, такой, чтобы отношение числа закодированных кубитов к их общему количеству приближалось к единице.

8. Выявление квантовых утечек

Сейчас я хотел бы более подробно рассмотреть одно из предположений, сделанных в предыдущем анализе. Мы проигнорировали вероятность *утечки*. В нашей модели квантового компьютера каждый из кубитов живет в двумерном гильбертовом пространстве. Мы предположили, что при возникновении ошибки этот кубит либо запутывается с окружением, либо поворачивается в двумерном пространстве в непредсказуемом направлении. Но существует другой возможный тип ошибки, когда кубит из двумерного просачивается в более широкое пространство (Plenio & Knight 1996). Например, в компьютере на ионных ловушках квантовую информацию можно хранить в двумерном пространстве, натянутом на основное состояние иона и некоторое долгоживущее метастабильное состояние (Cirac & Zoller 1995). Но в процессе работы устройства ион может совершить неожиданный переход в другое состояние. Если это состояние быстро распадается в основное состояние, то ошибка может быть обнаружена и исправлена с помощью стандартных на данный момент методов отказоустойчивого исправления квантовых ошибок. Но если ион остается подвешенным в ложном пространстве в течение длительного времени, то эти методы приведут к сбою.

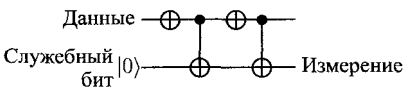


Рис. 16. Схема детектирования квантовой утечки. Если утечка произошла, то результат измерения равен $|0\rangle$, в противном случае — $|1\rangle$

Одна из стратегий решения этой проблемы состоит в определении наиболее вероятных уровней утечки и периодическое возвращение их в основное состояние, но при большом количестве таких уровней этот метод может оказаться слишком громоздким. Более элегантным решением является обнаружение утечки, но без попытки точно диагностировать, что произошло с просочившимся кубитом. Например, в ионной ловушке можно сконструировать CNOT-вентиль, действующий тривиально, если управляющий бит не находится в двумерном пространстве, которому принадлежит кубит. Тогда мы можем выполнить изображенную на рисунке 16 последовательность вентилей. При утечке информационного кубита ничего не происходит, но в случае отсутствия всякой утечки инвертируется служебный бит. Затем мы измеряем служебный бит. Результат измерения $|1\rangle$ проецирует кубит на правильное гильбертово пространство, тогда как результат $|0\rangle$ проецирует кубит на просочившееся состояние.

Если возникла утечка, то поврежденный кубит отбрасывается[1] и заменяется новым кубитом в стандартном состоянии, скажем, в состоянии $|0\rangle$. (Если используется каскадное кодирование, диагностика утечки должна осуществляться только на самых нижних уровнях кодирования.) После этого выполняется обычное измерение синдрома, которое спроецирует кубит на такое состояние, что ошибку можно будет исправить с помощью простого унитарного преобразования. Поскольку еще до измерения синдрома известно, что поврежденный кубит находится в конкретной позиции внутри блока, можно применить модернизированную версию исправления ошибок, разработанную с целью диагностики и исправления ошибок, находящихся в известных позициях (Grassl *и др.* 1996).

9. Машина мечты

Вспомним некоторые важные допущения, сделанные нами при оценке порога безошибочности:

[1] Конечно, позднее, после возвращения его в основное состояние, этим кубитом можно пользоваться снова.

- **Случайные ошибки.** Мы предположили, что ошибки не имеют никакой систематической составляющей.[1] Это допущение позволяет складывать вероятности, а не их амплитуды, для оценки того, как с течением времени растет вероятность появления ошибки. Систематические ошибки будут накапливаться гораздо быстрее, и, следовательно, допустимая частота ошибок будет намного меньше. Иначе говоря, если порог безошибочности для случайных ошибок равен ϵ_0, то для максимально законспирированных систематических ошибок он будет иметь порядок ϵ_0^2. Моя позиция такова: (1) даже если наши аппаратные средства предрасположены к совершению ошибок с систематическими фазами, эти ошибки будут стремиться к взаимному уничтожению в процессе достаточно продолжительного вычисления (Obenland & Despain 1996ab; Miquel *et al* 1997), и (2) поскольку систематические ошибки в принципе можно понять и устранить, с фундаментальной точки зрения более важным является понимание ограничений, накладываемых на работу машины случайными ошибками.
- **Некоррелированные ошибки.** Мы предположили, что ошибки являются некоррелированными как в пространстве, так и во времени. Таким образом, когда мы говорим, что вероятность ошибки на один кубит равна $\epsilon \sim 10^{-5}$, мы фактически имеем в виду, что для любой пары кубитов вероятность их одновременного повреждения ошибками имеет порядок $\epsilon^2 \sim 10^{-10}$. Это сильное и важное предположение, поскольку при возникновении нескольких ошибок в одном и том же блоке кода наши схемы кодирования дают сбой. Будущие квантовые инженеры столкнутся с проблемой обеспечения того, чтобы это предположение достаточно хорошо работало в конструируемых ими устройствах.
- **Максимальный параллелизм.** Мы предположили, что многие квантовые вентили могут выполняться параллельно на каждом такте вычислений. Это допущение дает возможность одновременного осуществления исправления ошибок во всех кодовых блоках, поэтому оно важно для контроля ошибок хранения кубитов. (Другими словами, добавление к коду следующего уровня каскадирования привело бы к увеличению вероятности сбоя, поскольку каждому не занятому в процессе отдельному кубиту пришлось бы дольше ожидать своей очереди исправления ошибок.) Если мы игнорируем ошибки запоминающего устройства, то в анализе порога безошибочности параллельная работа не является необходимой, но она, конечно, желательна для ускорения процесса вычисления.

[1] Нилл и др. (Knill *et al.* 1996, 1997) доказали существование порога безошибочности для гораздо более общих моделей ошибки.

- **Независимая от количества кубитов частота возникновения ошибок.** Мы предположили, что частота ошибок не зависит от количества хранящихся в устройстве кубитов. Неявно это предположение относится к природе аппаратных средств. Например, оно было бы необоснованным, если бы все кубиты хранились в единственной ионной ловушке и делили бы один фононный канал передачи информации. (Cirac & Zoller 1995).
- **Вентили могут действовать на любую пару кубитов.** Мы предположили, что наша машина оборудована набором фундаментальных вентилей, которые могут применяться к любой паре хранящихся кубитов (или к тройке кубитов в случае вентиля Тоффоли), независимо от степени их близости друг к другу. На практике возможны издержки как по времени выполнения, так и по частоте появления ошибок, связанные с перемещением кубитов для того, чтобы вентиль мог действовать эффективно на определенную пару. Оставим проблему выбора архитектуры, минимизирующей эти затраты, будущим конструкторам машин.
- **Новые служебные кубиты.** Мы предположили, что наш компьютер имеет доступ к достаточному запасу свежих служебных кубитов. Служебные кубиты используются как для выполнения вентилей (Тоффоли), так и для осуществления исправления ошибок. С накоплением эффектов случайных ошибок генерируется энтропия, а процесс исправления ошибок выбрасывает ее из вычислительного устройства в служебные регистры. В принципе, вычисление может продолжаться независимо по мере поступления новых служебных кубитов, но на практике мы захотим очистить служебный кубит и использовать его снова. Стирание служебного кубита неизбежно вызовет рассеяние мощности и образование тепла; таким образом, потребуется охлаждение устройства.
- **Никаких ошибок утечки.** Ошибки утечки игнорировались. Но, как отмечено в разделе 8, их учет не существенно влияет на выводы.

Наши предположения были достаточно реалистичными, поэтому можно смело утверждать, что квантовый компьютер с количеством кубитов порядка миллиона и частотой ошибок на вентиль порядка одной на миллион будет мощным и полезным устройством (при условии достаточной скорости обработки данных). С точки зрения текущего состояния технологии (Monroe *et al* 1995; Turchette *et al* 1995; Cory *et al* 1996; Gershenfeld & Chuang 1997), эти числа выглядят пугающе. Но на самом деле даже машина, удовлетворяющая гораздо менее жестким техническим требованиям, все же может быть очень полезна (Preskill 1997). Прежде всего, поми-

мо факторизации, квантовые компьютеры могут выполнять другие задачи, и некоторые из них (в частности, моделирование квантовых систем (Lloyd 1996)) могут быть совершены менее надежным или менее крупным устройством. Более того, по ряду причин наша оценка порога безошибочности может оказаться слишком консервативной. Например, она была получена при допущении, что фазовые и амплитудные ошибки в кубитах одинаково вероятны. Располагая более реалистичной моделью, лучше описывающей вероятности ошибок в действующем устройстве, можно было бы осуществить более эффективную схему исправления ошибок, позволяющую пережить более высокую частоту их появления. Кроме того, даже в сформулированных предположениях приведенный выше анализ отказоустойчивой схемы не вполне точен; при более тонком анализе можно ожидать несколько более высокого порога безошибочности, возможно, даже значительно более высокого. Существенного улучшения можно добиться и путем модификации отказоустойчивых схем либо в результате обнаружения более эффективного способа реализации универсального набора отказоустойчивых вентилей, либо в результате открытия более эффективных методов выполнения измерения синдрома ошибки. Я не буду удивлен, если окажется, что квантовый компьютер со всевозможными усовершенствованиями может работать эффективно с вероятностью ошибки на вентиль, скажем, порядка 10^{-4} (число 10^{-4} вполне может оказаться *ниже* порога безошибочности; более оптимистичные оценки порога безошибочности были выдвинуты Залкой (Zalka 1996)).

В любом случае, принимая эти оценки в качестве номинальных, мы получаем приблизительную цель, к которой должны стремиться: 10^6 кубитов с частотой ошибок 10^{-6}. Это звучит достаточно жестко, но, конечно, могло быть и хуже. Если бы мы пришли к выводу, что нам необходима частота ошибок, скажем, порядка 10^{-20}, тогда будущие перспективы квантовых вычислений были бы действительно неясными. Частота ошибок 10^{-6} несомненно претенциозна, но, возможно, не находится за рамками того, что может быть достигнуто в будущем. В любом случае, сейчас мы имеем четкое представление о том, насколько хорошей должна быть работа квантового компьютера. И это по сути уже является небывалым прогрессом по сравнению с прошлым годом.

Эта работа выполнена при частичной поддержке Департамента Энергетики, грант DE-FG03-92-ER40701, а также Управления перспективного планирования оборонных научно-исследовательских работ (DARPA), грант DAAH04-96-1-0386 управляемый Военным исследовательским центром. Я признателен Дэвиду ДиВинченцо и Войцеху Зуреку за организа-

цию стимулирующих встреч, я благодарю Эндрю Стина и Кристофа Залку за полезные замечания к рукописи этой статьи. Я также хочу поблагодарить своих сотрудников Дэвида Бэкмана, Джара Эвслина, Шэма Какадэ и особенно Дэниела Готтесмана за многочисленные продуктивные дискуссии по проблемам отказоустойчивых вычислений.

Литература

Aharonov D. & Ben-Or M. 1996a Fault tolerant quantum computation with constant error. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* ACM. New York, 1998, p. 176. (Online preprint quant-ph/9611025.)

Beckman D., Chari A., Devabhaktuni S. & Preskill J. 1996 Efficient networks for quantum factoring. *Phys. Rev.* A **54**, 1034–1063.

Bennett C., DiVincenzo D., Smolin J. & Wootters W. 1996 Mixed state entanglement and quantum error correction. *Phys. Rev.* A **54**, 3824–3851.

Calderbank A. R. & Shor P. W. 1996 Good quantum error-correcting codes exist. *Phys. Rev.* A **54**, 1098–1105.

Calderbank A. R., Rains E. M., Shor P. W. & Sloane N. J. A. 1996 Quantum error correction via codes over GF(4). IEEE Trans. Inf. Theory, 44(4), pp. 1369–1387 (Online preprint quant-ph/9608006.)

Calderbank A. R., Rains E. M., Shor P. W. & Sloane N. J. A. 1997 Quantum error correction and orthogonal geometry. *Phys. Rev. Lett.* **78**, 405–408.

Cirac J. I. & Zoller P. 1995 Quantum computations with cold trapped ions. *Phys. Rev. Lett.* **74**, 4091–4094.

Cory D. G., Fahmy A. F. & Havel T. F. 1996 Nuclear magnetic resonance spectroscopy: an experimentally accessible paradigm for quantum computing. In *Proceedings of the 4th Workshop on Physics and Computation*, T. Toffoli, M. Biafore, and J. Leao (eds.), Boston: New England Complex Systems Institute, pp. 87–91.

Dieks D. 1982 Communication by electron-paramagnetic-resonance devices. *Phys. Lett.* A **92**, 271–272.

DiVincenzo D. & Shor P. 1996 Fault-tolerant error correction with efficient quantum codes. *Phys. Rev. Lett.* **77**, 3260–3263.

Gershenfeld N. & Chuang I. 1997 Bulk spin resonance quantum computation. *Science* **275**, 350–356.

Griffiths R. B. & Niu C. 1996 Semiclassical Fourier transform for quantum computation. *Phys. Rev. Lett.* **76**, 3228–3231.

Gottesman D. 1996 Class of quantum error-correcting codes saturating the quantum Hamming bound. *Phys. Rev.* A **54**, 1862–1868.

Gottesman D. 1997a A theory of fault-tolerant quantum computation. *Phys. Rev.* A **57**, 127–137 (1998). (Online preprint quant-ph/9702029.)

Gottesman D. 1997b Stabilizer codes and quantum error correction. Ph. D. thesis, California Institute of Technology. (Online preprint quant-ph/9705052.)

Gottesman D., Evslin J. Kakade S. & Preskill J. 1996, to be published.

Grassl M., Beth Th. & Pellizzari T. 1996 Codes for the quantum erasure channel. *Phys. Rev.* A **56**, 33–38 (1997). (Online preprint quant-ph/9610042.)

Hughes R. 1997 (private communication).

Kitaev A. Yu. 1996a Quantum error correction with imperfect gates, in *Proceedings of the Third International Conference on Quantum Communication and Measurement*, Ed O. Hirota, A/S/ Holevo, and C.M. Caves, pp. 181–188 (New York, Plenum, 1997).

Kitaev A. Yu. 1996b. Китаев А.Ю. Квантовые вычисления: алгоритмы и исправления ошибок, Успехи мат. наук, **52**, стр. 53–112 (1997).

Knill E. & Laflamme R. 1996 Concatenated quantum codes. Techn. Report LAOR-96-2808. (Online preprint quant-ph/9608012.)

Knill E. & Laflamme R. 1997. A theory of quantum error-correcting codes. *Phys. Rev.* A **55**, 900–911.

Knill E., Laflamme R. & Zurek W. 1996 Accuracy threshold for quantum computation. (Online preprint quant-ph/9610011.)

Knill E., Laflamme R. & Zurek W. 1997 Resilient quantum computation: error models and thresholds. *Proc. Roy. Soc. Lond.* A **454**, 365–384 (1998) (Online preprint quant-ph/9702058.)

Laflamme R., Miquel C., Paz J. P., & Zurek W. 1996 Perfect quantum error correction code. *Phys. Rev. Lett.* **77**, 198–201.

Landauer R. 1995 Is quantum mechanics useful? *Phil. Tran. R. Soc. Lond.* **353**, 367–376.

Landauer R. 1996 The physical nature of information. *Phys. Lett.* A **217**, 188–193.

Landauer R. 1997 Is quantum mechanically coherent computation useful? In *Proc. Drexel-4 Symposium on Quantum Nonintegrability-Quantum-Classical Correspondence*, Philadelphia, PA, 8 September 1994 (ed. D. H. Feng and B.-L. Hu), Boston: International Press.

Lenstra A. K., Cowie J., Elkenbracht-Huizing M., Furmanski W., Montgome-

ry P. L., Weber D. & Zayer J. 1996 RSA factoring-by-web: the world-wide status. (Online document http://www.npac.syr.edu/factoring/status.html.)

Lloyd S. 1996 Universal quantum simulators. *Science* **273**, 1073–1078; correction in *Science* **279**, 1113–1117 (1998).

Lloyd, S. 1997 The capacity of a noisy quantum channel. *Phys. Rev.* A **55**, 1613–1622.

MacWilliams F. J. & Sloane N. J. A. 1977 *The Theory of Error-Correcting Codes*. New York: North-Holland Publishing Company. Русский перевод: Ф.Дж. Мак-Вильямс, Н.Дж. Слоэн, *Теория кодов, исправляющих ошибки*, Связь, М., 1979.

Miquel C., Paz J. P. & Zurek W. H. 1997 Quantum computation with phase drift errors, *Phys. Rev. Lett.*, **78**, 3971–3974 (1997); (Online preprint quant-ph/9704003.)

Monroe C., Meekhof D. M., King B. E., Itano W. M. & Wineland D. J. 1995 Demonstration of a fundamental quantum logic gate. *Phys. Rev. Lett.* **75**, 4714–4717.

Obenland K. & Despain A. M. 1996a Simulation of factoring on a quantum computer architecture. In *Proceedings of the 4th Workshop on Physics and Computation*, Boston, November 22–24, 1996, Boston: New England Complex Systems Institute.

Obenland K. & Despain A. M. 1996b Impact of errors on a quantum computer architecture. Technical Report, Information Science Institute, University of Southhern California, Oct 1, 1996;
(online preprint http:// www.isi.eu/acal/quantum/quantumoperrors.ps, 1996).

Plenio M. B. & Knight P. L. 1996 Decoherence limits to quantum computation using trapped ions. *Proc. Roy. Soc. Lond.* A **453**, 2017–2041 (1997) (Online preprint quant-ph/9610015.)

Preskill J. 1997 Quantum computing: pro and con. *Proc. Roy. Soc. Lond.* A **454**, 469–486 (1998) (Online preprint quant-ph/9705032.); перевод в *Квантовые вычисления: за и против*. — РХД, Ижевск (1999).

Shor P. 1994 Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Fundamentals of Computer Science*. Los Alamitos, CA: IEEE Press, pp. 124–134. Расширенная версия: SIAM J. Comp. **26**, 1484–1509 (1997), online preprint quant-ph/9508027; перевод: П. Шор, Полиномиальные по времени алгоритмы разложения числа на простые множители и нахождения дискретного логарифма для квантовых компьютеров. *Квантовый компьютер и квантовые вычисления*. — Ижевск, РХД (1999).

Shor P. 1995 Scheme for reducing decoherence in quantum memory. *Phys. Rev.* A **52**, R2493–R2496.

Shor P. 1996 Fault-tolerant quantum computation. In *Proceedings of 37 Annual Symposium on the Foundations of Computer Science.* pp. 56–65, Los Alamitos, CA: IEEE Press (Online preprint quant-ph/9605011).

Shor P. 1997, these proceedings.

Shor P. & and Smolin J. 1996 Quantum error-correcting codes need not completely reveal the error syndrome. (Online preprint quant-ph/9604006.)

Steane A. M. 1996a Error correcting codes in quantum theory. *Phys. Rev. Lett.* **77**, 793–797.

Steane A. M. 1996b Multiparticle interference and quantum error correction. *Proc. Roy. Soc. Lond.* A **452**, 2551–2577.

Steane A. M. 1997 Active stabilization, quantum computation and quantum state synthesis. *Phys. Rev. Lett.* **78**, 2252–2255.

Turchette Q. A., Hood C. J., Lange W., Mabuchi H. & Kimble H. J. 1995 Measurement of conditional phase shifts for quantum logic. *Phys. Rev. Lett.* **75**, 4710–4713 (1995).

Vedral V., Barenco A. & Ekert A. 1996 Quantum networks for elementary arithmetic operations. *Phys. Rev.* A **54**, 147–153

Unruh W. G. 1995 Maintaining coherence in quantum computers. *Phys. Rev.* A **51**, 992–997.

Wootters W. K. & Zurek W. H. 1982 A single quantum cannot be cloned. *Nature* **299**, 802–803; *Nature* **304**, 188-189 (1983).

Zalka C. 1996 Threshold estimate for fault tolerant quantum computing. (Online preprint quantph/9612028.)

Отказоустойчивые квантовые вычисления[1]

Джон Прескилл[2]

Открытие *коррекции квантовых ошибок* существенно улучшило долгосрочные перспективы технологии квантовых вычислений. Закодированную квантовую информацию можно защитить от ошибок, возникающих вследствие неконтролируемых взаимодействий с окружением или неидельного выполнения квантовых логических операций. Исправление ошибок может быть эффективным даже при возникновении случайных ошибок в ходе самой процедуры восстановления. Более того, закодированную квантовую информацию можно обрабатывать без серьезного распространения ошибок. В принципе, сколь угодно продолжительное квантовое вычисление может быть надежно выполнено при условии, что средняя вероятность появления ошибки на квантовый вентиль меньше определенной критической величины, называемой *порогом безошибочности*. Возможна разработка средств аппаратного обеспечения квантовых вычислений, обладающих собственной внутренней отказоустойчивостью, с привлечением топологических взаимодействий Аронова–Бома для обработки квантовой информации.

1. Потребность в отказоустойчивости

Квантовые компьютеры кажутся способными, во всяком случае, в принципе, решать определенные задачи намного быстрее, чем любой мыслимый классический компьютер [1–3]. Однако на практике технология квантовых вычислений все еще находится в зачаточном состоянии. Несмотря на то, что практичный и полезный квантовый компьютер, возможно, будет сконструирован уже в обозримом будущем, мы до сих пор не можем ясно представить, как будут выглядеть аппаратные средства этой машины. Тем не менее, мы можем быть абсолютно уверены в том, что любой практичный квантовый компьютер будет включать в свою работу определенный тип

[1] J. Preskill, Fault-tolerant Quantum Computation, *Introduction to Quantum Computation*, H.-K. Lo, S. Popesku, T. P. Spiller (Eds.), World Scientific, Singapore et al. (1998).

[2]Калифорнийский технологический институт, Пасадена, CA 91125, США.

исправления ошибок. По сравнению с обычными цифровыми, квантовые компьютеры гораздо более подвержены ошибкам, поэтому для предотвращения аварийных отказов в их работе потребуются определенные методы контроля и исправления этих ошибок.

Декогерентизация — самый грозный враг квантового компьютера [4–8]. Мы знаем, как приготовить квантовое кот-состояние, то есть суперпозицию живого и мертвого кота, но мы никогда не сможем наблюдать такие макроскопические суперпозиции вследствие их крайней нестабильности. Ни один реально существующий кот не может быть изолирован от окружающей его среды: окружение «измеряет» кота, незамедлительно проецируя его на состояние — совершенно живой или совершенно мертвый [9]. Квантовый компьютер, возможно, не так сложен, как кот, но это сложная квантовая система, и, подобно коту, она взаимодействует с окружением. Хранящаяся в компьютере информация разрушается, что ведет к ошибкам и сбою в вычислениях. Можем ли мы защитить квантовый компьютер от подрывающего влияния декогерентизации?

Декогерентизация — не единственный наш враг [4–6]. Даже если бы мы были способны добиться превосходной изоляции нашего компьютера от окружающей среды, мы не могли бы рассчитывать на безупречно точное выполнение квантовых логических вентилей. Как и в классическом аналоговом компьютере, ошибки в квантовых вентилях образуют континуум. В ходе вычисления мелкие ошибки в вентилях могут накапливаться и, в конечном счете, вызвать сбой, а способ их исправления неочевиден. Можем ли мы предотвратить катастрофическое накопление этих малых ошибок?

Перспективы на будущее квантовых вычислений получили огромную поддержку благодаря открытию, что коррекция квантовых ошибок, в принципе, действительно возможна [10–12]. Но самого по себе этого открытия недостаточно, чтобы гарантировать надежную работу шумящего квантового компьютера. Чтобы выполнить протокол квантовой коррекции ошибок, мы должны сначала закодировать квантовую информацию, которую хотим защитить, а затем многократно выполнить операции по исправлению, которые обращают накапливаемые ошибки. Но кодирование и исправление сами по себе являются сложными квантовыми вычислениями, и в ходе их выполнения неизбежно будут возникать новые ошибки. Следовательно, чтобы достичь высокой надежности даже при появлении некоторых ошибок на этапе восстановления, нам необходимо найти достаточно сильные методы их исправления.

Более того, чтобы оперировать квантовым компьютером, мы должны уметь больше, чем просто *хранить* квантовую информацию; мы должны уметь *обрабатывать* ее. Мы должны быть в состоянии выполнять кван-

товые вентили, в которых два или более закодированных кубитов взаимодействуют друг с другом. Если в одном кубите возникает ошибка, то существует вероятность ее распространения на другие кубиты, с которыми зараженный кубит будет взаимодействовать в ходе выполняемых вслед за этим квантовых вентилей. Мы должны сконструировать наши вентили таким образом, чтобы минимизировать распространение ошибок.

Включение коррекции квантовых ошибок несомненно усложнит работу квантового компьютера. Создание необходимого для защиты от ошибок резерва потребует увеличить количество элементарных кубитов. Применение вентилей к закодированной информации и периодическое включение этапов исправления ошибок замедлит процесс вычисления. Ввиду этого необходимого усложнения устройства *a priori* не очевидно, что коррекция ошибок приведет к улучшению его работы.

Устройство, работающее эффективно, даже если его элементарные компоненты неидеальны, называют *отказоустойчивым*. Настоящий раздел посвящается теории отказоустойчивых квантовых вычислений. В ней мы исследуем сформулированные выше проблемы и вопросы.

В самом деле, подобные вопросы возникают и в теории отказоустойчивых *классических* вычислений. Но поскольку существующая, базирующаяся на кремнии, полупроводниковая схемотехника в высшей степени надежна, отказоустойчивость не является важнейшим условием работоспособности современных цифровых компьютеров. Несмотря на это, изучение отказоустойчивых классических вычислений имеет выдающуюся историю. В 1952 году фон Нейман [13] предложил повысить надежность работы схемы с шумящими вентилями путем многократного выполнения каждого вентиля и применения мажоритарной схемы. Он сделал вывод, что если отказы вентилей статистически независимы, а вероятность отказа в расчете на один вентиль достаточно мала, то любое вычисление можно выполнить с достаточной надежностью. Единственным недостатком анализа фон Неймана было то, что он предположил идеальную передачу битов по соединяющим вентили «проводникам».[1] Выход за рамки этого предположения оказался сложным, но, в конечном счете, успех был достигнут в 1983 году Гаксом [14], который описал универсальный клеточный автомат с иерархической структурой, которая может поддерживаться локальными операциями в присутствии шума, не нуждаясь в непосредственной нелокальной связи между его компонентами.

[1]Это достаточно серьезная проблема, ибо схема фон Неймана не может быть реализована в трехмерном пространстве с ограниченными по длине проводниками; предполагается, что вероятность появления ошибки передачи стремится к единице при стремящейся к бесконечности длине линий передачи.

Интересен вопрос: способна ли квантовая система подобным образом обеспечивать функционирование сложной иерархической структуры? Однако мы не будем столь амбициозны, чтобы браться здесь за эту проблему. Поскольку нас интересуют ограничения, накладываемые шумом на обработку *квантовой* информации, мы разделим наши вентили на классические и квантовые и будем считать, что классические вентили могут выполняться с идеальной точностью и так быстро, как это необходимо.[1] Это предположение будет хорошо обоснованным до тех пор, пока тактовая частота и точность классического компьютера значительно превосходят соответствующие характеристики квантового компьютера.

Обсудив во втором разделе свойства конкретного корректирующего кода (семикубитовый код Стина [12]), в следующем разделе мы сделаем краткий обзор основных принципов отказоустойчивого исправления квантовых ошибок. Ошибки, возникающие в ходе самого процесса исправления, впоследствии могут повредить закодированную квантовую информацию; следовательно, эффективность коррекции требует особой тщательности ее выполнения. Для измерения *синдромов*, диагностирующих ошибки в блоках закодированных данных, используются служебные кубиты. Последние могут содержать ошибки, поэтому необходимо минимизировать вероятность их распространения на закодированную информацию. В третьем разделе описываются предложенные Питером Шором [13] и Эндрю Стином [16] методы управления распространением ошибок в процессе исправления.

Предмет четвертого раздела составляет отказоустойчивая обработка квантовой информации. Здесь главная проблема заключается в создании универсального набора квантовых вентилей, способных действовать на блоки закодированных данных, не внося непомерного количества ошибок. В этом разделе сделан обзор некоторых схем универсальных вычислений (Питер Шор [15] и Дэниел Готтесман [17]).

Коль скоро элементарные вентили нашего квантового компьютера достаточно надежны, мы можем применять их к закодированной информации, а также в процессах отказоустойчивого исправления ошибок, повышая таким образом надежность нашего устройства. Но для любого заданного квантового кода или даже большинства бесконечных классов, содержащих коды с блоками сколь угодно больших размеров, при выполнении достаточно продолжительного вычисления рано или поздно эти процедуры дадут сбой. Однако, как показано в разделе 5, существует особый класс кодов (*каскадные коды*), позволяющих надежно выполнять все более и бо-

[1] Однако, когда мы будем рассматривать коррекцию ошибок с помощью кодов со сколь угодно большими размерами блоков, мы будем требовать, чтобы объем выполняемых классических операций был полиномиально ограничен по размеру блоков.

лее продолжительные квантовые вычисления, требуя для этого умеренной скорости увеличения размера блока [18–24]. Применяя каскадные коды, мы можем установить *порог безошибочности* квантовых вычислений; как только наше «железо» будет удовлетворять установленным критерием точности, корректирующие ошибки квантовые коды и отказоустойчивые процедуры позволят осуществлять сколь угодно длинные квантовые вычисления со сколь угодно высокой надежностью. Этот результат приблизительно аналогичен выводу фон Неймана относительно классической отказоустойчивости, тогда как иерархическая структура каскадного кодирования напоминает конструкцию Гакса. Приводится набросок оценки порога безошибочности в предположениях о характере ошибок, перечисленных в разделе 6.

С развитием отказоустойчивых методов стало ясно, что, активно вмешиваясь в работу квантового компьютера, оператор, в принципе, может защитить его от ошибок в шумящем (но не *слишком сильно*) окружении. Хотя, в более далекой перспективе, надежность практичного квантового компьютера может быть достигнута совершенно иным способом — с помощью внутренне отказоустойчивого аппаратного обеспечения. Такое «железо», устойчивое к локальным возмущениям, могло бы оперировать сравнительно небрежно организованными процедурами и, тем не менее, надежно хранить и обрабатывать квантовую информацию. Предмет раздела 7 составляет предложенный Алексеем Китаевым [25] проект отказоустойчивых аппаратных средств, в которых квантовые вентили используют неабелевы взаимодействия Аароно́ва–Бо́ма между пространственно разделенными квазичастицами в гипотетически возможной двумерной спиновой системе. Хотя лабораторная реализация идеи Китаева, возможно, является делом отдаленного будущего, его работа открывает новый подход к проблеме квантовой отказоустойчивости, позволяющий отказаться от анализа абстрактных квантовых схем в пользу поиска новых физических принципов, которые можно было бы использовать для надежной обработки квантовой информации.

С точки зрения современной технологии, сделанные в этой главе заявления относительно потенциально возможного отказоустойчивого управления сложными квантовыми состояниями могут показаться претенциозными. Конечно, нам предстоит еще долгий путь, прежде чем будут созданы устройства, которые смогут, скажем, использовать порог безошибочности квантовых вычислений. Тем не менее, я уверен в том, что сегодняшняя работа по коррекции квантовых ошибок будет иметь продолжение. Последние три года теория квантовых вычислений развивалась с захватывающей скоростью. Если квантовая классификация вычислительной сложности отличается от классической (что очень похоже на правду), то ни один мыслимый классический компьютер не сможет точно предсказать поведение

даже умеренного числа кубитов (порядка 100). Тогда, возможно, относительно маленькие квантовые системы будут иметь намного больший потенциал, чем тот, что мы можем сегодня представить. Этот потенциал до сих пор не реализован, так как пока мы не в состоянии защитить такие системы от разрушительного влияния шума и декогерентизации. Таким образом, открытие отказоустойчивых методов коррекции квантовых ошибок и квантовых вычислений имеет исключительно глубокие последствия, как для будущего экспериментальной физики, так и для будущего технологии. Теоретические достижения осветили дорогу в будущее, в котором сложные квантовые системы, вероятно, окажутся более покладистыми.

2. Коррекция квантовых ошибок: 7-кубитовый код

Чтобы понять, почему возможна коррекция квантовых ошибок, очень поучительно изучить конкретный код. Простым и важным примером кода, исправляющего квантовые ошибки, является 7-кубитовый код, разработанный Эндрю Стином [11,12]. Этот код позволяет хранить один кубит квантовой информации (произвольное состояние в двумерном гильбертовом пространстве), используя всего семь кубитов (погружая двумерное гильбертово пространство в пространство размерности 2^7). Фактически код Стина тесно связан с известным классическим корректирующим ошибки [7,4,3]-кодом Хэмминга [26]. Чтобы понять принцип работы кода Стина, важно сначала понять классический код Хэмминга.

Код Хэмминга использует блок из семи битов для кодирования четырех битов классической информации: имеется $16 = 2^4$ строк длины семь, представляющих собой истинные кодовые слова. Кодовые слова можно характеризовать матрицей проверки четности:

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{1}$$

Каждое истинное кодовое слово представляет собой 7-битовый вектор (-столбец) v_{code}, удовлетворяющий уравнению

$$\sum_k H_{jk}(v_{\text{code}})_k = 0 \pmod 2; \tag{2}$$

то есть в арифметике по модулю 2 матрица H уничтожает каждое кодовое слово. Поскольку $Z_2 = \{0, 1\}$ является (конечным) полем, в нем применимы результаты линейной алгебры. H имеет три линейно независимых стро-

ки, а ее ядро[1] натянуто на четыре линейно независимых вектора-столбца. 16 истинных кодовых слов представляют собой все возможные линейные комбинации этих четырех векторов с коэффициентами, выбираемыми из $\{0, 1\}$.

Теперь предположим, что v_{code} — (неизвестное) истинное кодовое слово, в котором возникла единственная (неизвестная) ошибка: один из семи битов инвертировался. Наша задача — определить поврежденный бит и исправить ошибку. Этот трюк можно выполнить с помощью матрицы проверки четности. Пусть e_i обозначает вектор с единицей в i-ой позиции и нулями в остальных. Тогда при инвертировании i-го бита v_{code} становится равным $v_{\text{code}} + e_i$. Если мы подействуем на этот вектор матрицей H, то получим

$$H(v_{\text{code}} + e_i) = He_i \tag{3}$$

(поскольку матрица H уничтожает кодовое слово v_{code}), что является именно i-м столбцом матрицы H. Поскольку все столбцы матрицы H различны, то этим однозначно определяется значение i. Выяснив местоположение ошибки, ее можно исправить путем повторного инвертирования i-го бита. Таким образом, если инвертируется только один бит, то мы можем однозначно восстановить закодированную информацию; но при инвертировании двух или большего числа разных битов закодированные данные будут повреждены. Примечательно то, что величина He_i выявляет позицию ошибки, ничего не сообщая о кодовом слове v_{code}, то есть не раскрывая закодированную информацию.

Код Стина обобщает этот вид классического кода коррекции ошибок на *квантовый* код, корректирующий ошибки. Он использует 7-кубитовый «блок» для кодирования одного кубита квантовой информации, то есть произвольного состояния в двумерном гильбертовом пространстве, натянутом на два состояния: «логический нуль» $|0\rangle_{\text{code}}$ и «логическая единица» $|1\rangle_{\text{code}}$. Код сконструирован так, что он позволяет исправить одну произвольную ошибку, возникающую в любом из семи кубитов в блоке.

Что мы подразумеваем под произвольной ошибкой? Рассматриваемый кубит может подвергнуться случайному *унитарному* преобразованию или *потерять когерентность*, запутавшись с состояниями окружающей среды. Предположим, что неповрежденный кубит должен находиться в состоянии $a|0\rangle + b|1\rangle$. (Конечно, он может быть запутан с остальными, так что коэффициенты a и b не обязательно являются комплексными числами; это могут быть состояния, одновременно ортогональные $|0\rangle$ и $|1\rangle$, которые, как мы

[1] Ядро матрицы или ее *нуль-пространство* — пространство векторов, удовлетворяющих уравнению (2). — *Прим. ред.*

предполагаем, не подвержены ошибкам.) Теперь, если этот кубит поврежден произвольной ошибкой, результирующее состояние можно разложить следующим образом:

$$\begin{aligned} a|0\rangle + b|1\rangle \rightarrow & (a|0\rangle + b|1\rangle) \otimes |A_{\text{no error}}\rangle_{\text{env}} + \\ & + (a|1\rangle + b|0\rangle) \otimes |A_{\text{bit-flip}}\rangle_{\text{env}} + \\ & + (a|0\rangle - b|1\rangle) \otimes |A_{\text{phase-flip}}\rangle_{\text{env}} + \\ & + (a|1\rangle - b|0\rangle) \otimes |A_{\text{both error}}\rangle_{\text{env}}, \end{aligned} \tag{4}$$

где каждое $|A\rangle_{\text{env}}$ обозначает состояние окружающей среды. Мы не делаем каких-то определенных предположений относительно ортогональности или нормировки состояний $|A\rangle_{\text{env}}$,[1] поэтому уравнение (4) не влечет за собой потери общности. Мы приходим к выводу, что эволюцию кубита можно выразить как линейную комбинацию четырех возможностей: (1) не возникает никакой ошибки, (2) возникает инвертирование бита $|0\rangle \longleftrightarrow |1\rangle$, (3) возникает обращение относительной фазы $|0\rangle$ и $|1\rangle$, (4) происходит одновременное инвертирование бита и обращение фазы.

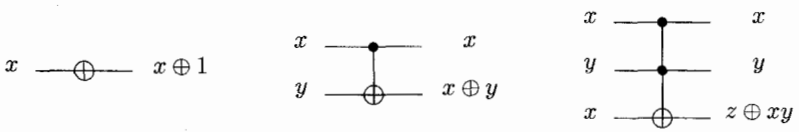


Рис. 1. Схематическое изображение NOT-вентиля, XOR-вентиля (контролируемое NOT) и вентиля Тоффоли (дважды контролируемое NOT)

Теперь понятно, как должен работать квантовый код, корректирующий ошибки [12, 27]. Выполняя подходящее измерение, мы хотим диагностировать, какая из этих четырех возможностей фактически осуществилась. Конечно, в общем случае состоянием кубита будет линейная комбинация этих четырех состояний, но измерение должно спроецировать его на базис, используемый в уравнении (4). После этого мы можем приступить к исправлению ошибки с помощью одного из четырех унитарных преобразований:

$$(1)\quad I, \quad (2)\quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (3)\quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (4)\quad Y = X \cdot Z \tag{5}$$

(какое из них применить, указывает результат измерения). Применив это преобразование, мы возвращаем кубит в его исходное состояние и полностью распутываем квантовые состояния кубита и окружающей среды.

[1] Хотя, конечно, совместная эволюция кубита и окружения должна быть унитарной.

Существенно, что при диагностике ошибки мы ничего не узнаем о закодированной квантовой информации, поскольку получение любой информации о коэффициентах a и b в уравнении (4) неизбежно приведет к разрушению когерентности кубита.

Если мы используем код Стина, отвечающее этим критериям измерение возможно. Логический нуль является равновзвешенной суперпозицией всех кодовых слов кода Хэмминга (H) с четным весом (слов с четным количеством единиц),

$$\begin{aligned}|0\rangle_{\text{code}} &= \frac{1}{\sqrt{8}} \sum_{v_{\text{even}} \in \text{H}} |v_{\text{even}}\rangle = \\ &= \frac{1}{\sqrt{8}} (|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle + \\ &+ |1010101\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle),\end{aligned} \tag{6}$$

а логическая единица является равновзвешенной суперпозицией всех кодовых слов кода Хэмминга с нечетным весом (слов с нечетным количеством единиц),

$$\begin{aligned}|1\rangle_{\text{code}} &= \frac{1}{\sqrt{8}} \sum_{v_{\text{even}} \in \text{H}} |v_{\text{even}}\rangle = \\ &= \frac{1}{\sqrt{8}} (|1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle + \\ &+ |0101010\rangle + |0100101\rangle + |0011001\rangle + |0010110\rangle).\end{aligned} \tag{7}$$

Поскольку все фигурирующие в уравнениях (6) и (7) состояния являются кодовыми словами Хэмминга, инвертирование единственного бита в блоке несложно обнаружить, выполняя простое квантовое вычисление, как это показано на рисунке 2 (используются приведенные на рис. 1 обозначения). Мы расширяем блок из семи кубитов тремя служебными кубитами[1] и выполняем унитарную операцию:

$$|v\rangle \otimes |0\rangle_{\text{anc}} \rightarrow |v\rangle \otimes |Hv\rangle_{\text{anc}}, \tag{8}$$

где H — матрица проверки четности Хэмминга, а $|\cdot\rangle_{\text{anc}}$ обозначает состояние трех служебных битов. Если ошибка содержится лишь в одном из семи кубитов, то измерение служебного кубита спроецирует его либо на инвертированное состояние, либо на исходное неповрежденное (но не на любую

[1] Чтобы сделать процедуру отказоустойчивой, нам потребуется увеличить число служебных кубитов, как это обсуждается в разделе 3.

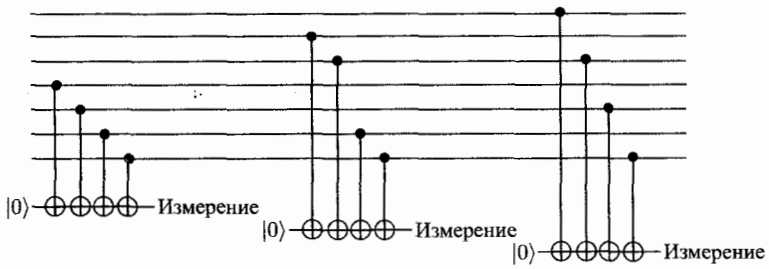


Рис. 2. Вычисление синдрома инвертирования бита для 7-кубитового кода Стина. Повторение вычисления в повернутом базисе диагностирует ошибки обращения фазы. Чтобы сделать процедуру отказоустойчивой, каждый служебный кубит нужно заменить четырьмя кубитами в надлежащем состоянии

нетривиальную суперпозицию этих двух состояний). В первом случае результат измерения диагностирует поврежденный бит, ничего не сообщая о закодированной в блоке квантовой информации.

Но чтобы осуществить коррекцию квантовых ошибок, нам понадобится наряду с ошибками инвертирования бита диагностировать и фазовые ошибки. Для достижения этой цели мы можем (следуя Стину [11, 12]) изменить базис для каждого кубита, применив поворот Адамара:

$$R = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{9}$$

Тогда ошибки, бывшие фазовыми в базисе $|0\rangle, |1\rangle$, в повернутом базисе

$$|\tilde{0}\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\tilde{1}\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{10}$$

становятся ошибками инвертирования бита. Следовательно, достаточно того, что наш код способен диагностировать ошибки инвертирования бита в этом повернутом базисе. Но если мы применяем поворот Адамара к каждому из семи кубитов, логический нуль и логическая единица Стина в повернутом базисе принимают вид

$$\begin{aligned} |\tilde{0}\rangle_{\text{code}} &= \frac{1}{4} \sum_{v \in H} |v\rangle = \frac{1}{\sqrt{2}}(|0\rangle_{\text{code}} + |1\rangle_{\text{code}}), \\ |\tilde{1}\rangle_{\text{code}} &= \frac{1}{4} \sum_{v \in H} (-1)^{\text{wt}(v)} |v\rangle = \frac{1}{\sqrt{2}}(|0\rangle_{\text{code}} - |1\rangle_{\text{code}}) \end{aligned} \tag{11}$$

(где $\mathrm{wt}(v)$ обозначает вес v). Ключевым моментом является то, что $|\tilde{0}\rangle_{\mathrm{code}}$ и $|\tilde{1}\rangle_{\mathrm{code}}$, подобно $|0\rangle_{\mathrm{code}}$ и $|1\rangle_{\mathrm{code}}$ являются суперпозициями кодовых слов Хэмминга. Следовательно, в повернутом базисе, как и в исходном, мы можем выполнить проверку четности Хэмминга для диагностики инвертирования битов, которое в исходном базисе представляло собой обращение фазы. При условии, что поврежден лишь один кубит, выполнение проверки четности в обоих базисах полностью диагностирует ошибку и дает возможность ее исправить.

В описанной выше схеме исправления ошибок я предполагал, что ошибка поражает только один кубит в блоке. Очевидно, что это допущение нереалистично; обычно все кубиты в определенной степени запутываются с окружением. Однако, как мы уже видели, процедура определения синдрома ошибки, как правило, проецирует каждый кубит на исходное неповрежденное состояние. Для каждого кубита появление ошибки происходит с отличной от нуля, но малой по предположению, вероятностью, которую мы будем обозначать ϵ. Сейчас мы сделаем очень важное предположение: ошибки, действующие на разные кубиты в одном блоке, полностью некоррелированы между собой. При таком допущении вероятность появления двух ошибок имеет порядок ϵ^2 и, при достаточно малой величине ϵ, гораздо меньше вероятности возникновения одной ошибки. Поэтому, с точностью порядка ϵ, мы можем уверенно ограничить наше внимание случаем, когда ошибка содержится максимум в одном кубите блока. (На самом деле, для этого вывода не требуется, чтобы действующие на разные кубиты ошибки были *полностью* некоррелированными. Если все кубиты подвергаются воздействию одного и того же слабого магнитного поля, так что вероятность инвертирования каждого из них равна ϵ, все будет в порядке, поскольку вероятность переворота спинов имеет порядок ϵ^2. Проблему вызвал бы возникающий с вероятностью порядка ϵ процесс, при котором происходит инвертирование двух спинов одновременно.)

Но в (маловероятном) случае, когда в одном и том же блоке кода возникает две ошибки, наша процедура исправления обычно дает сбой. Если в одном и том же блоке инвертируются два бита, то проверка четности Хэмминга неправильно диагностирует ошибку. Восстановление вернет квантовое состояние в кодовое подпространство, но в *закодированной* в блоке информации произойдет инвертирование бита

$$|0\rangle_{\mathrm{code}} \rightarrow |1\rangle_{\mathrm{code}}, \quad |1\rangle_{\mathrm{code}} \rightarrow |0\rangle_{\mathrm{code}}. \tag{12}$$

Аналогично, если в одном и том же блоке возникают две фазовые ошибки, то есть две ошибки инвертирования битов в повернутом базисе, то после

восстановления в нем произойдет инвертирование бита в повернутом базисе или обращение фазы в исходном базисе

$$|0\rangle_{\text{code}} \to |0\rangle_{\text{code}}, \quad |1\rangle_{\text{code}} \to -|1\rangle_{\text{code}} \tag{13}$$

(если один кубит в блоке содержит фазовую ошибку, а другой — ошибку инвертирования бита, то исправление будет успешным).

Таким образом, код Стина способен повысить надежность хранения квантовой информации. Предположим, мы хотим сохранить один кубит в неизвестном чистом состоянии $|\psi\rangle$. Из-за несовершенства запоминающего устройства состояние ρ_{out}, которое мы мы восстановим, будет иметь точность воспроизведения

$$F \equiv \langle\psi|\rho_{\text{out}}|\psi\rangle = 1 - \epsilon. \tag{14}$$

Но если мы храним кубит с использованием 7-кубитового блочного кода Стина, если каждый из семи кубитов сохраняется с точностью воспроизведения $F = 1 - \epsilon$, если ошибки кубитов некоррелированны; и, наконец, если мы можем безукоризненно выполнять коррекцию ошибок, кодирование и декодирование (подробнее об этом ниже), то закодированная информация может храниться с улучшенной точностью воспроизведения $F = 1 - O(\epsilon^2)$.

Кубит в неизвестном состоянии можно закодировать, используя изображенную на рисунке 3 схему. Принцип работы кодирующего устройства проще понять, если использовать альтернативное выражение для матрицы проверки четности Хэмминга,

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}. \tag{15}$$

(Эта форма матрицы H получается из (1) перестановкой столбцов, то есть всего лишь изменением нумерации битов в блоке.) Четный субкод кода Хэмминга фактически является пространством, натянутым на строки матрицы H; поэтому мы видим, что (в этом представлении H) первые три бита строки полностью характеризуют представленные в субкоде данные. Оставшиеся четыре бита — биты четности, обеспечивающие необходимую для защиты от ошибок избыточность. При кодировании неизвестного состояния $a|0\rangle + b|1\rangle$ кодирующее устройство сначала использует два XOR-вентиля, чтобы приготовить состояние $a|0000000\rangle + b|0000111\rangle$ — суперпозицию четного и нечетного кодовых слов Хэмминга. Оставшаяся часть схемы добавляет к этому состоянию $|0\rangle_{\text{code}}$: повороты Адамара (R) готовят равновзвешенную суперпозицию всех восьми возможных значений для

первых трех битов в блоке, а оставшиеся XOR-вентили включают предписанные матрицей H биты проверки четности.

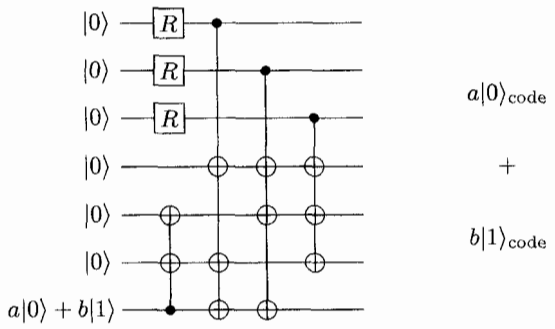


Рис. 3. Кодирующая схема для 7-кубитового кода Стина

Мы также хотим иметь возможность измерять закодированный кубит, скажем, проецируя его на ортогональный базис $\{|0\rangle_{\text{code}}, |1\rangle_{\text{code}}\}$. Если мы не возражаем против разрушения закодированного блока в процессе измерения, то достаточно измерить каждый из семи кубитов в блоке, проецируя на базис $\{|0\rangle, |1\rangle\}$; затем, чтобы получить кодовое слово Хэмминга, мы выполняем классическую коррекцию ошибок в результатах измерения. Четность этого кодового слова является значением логического кубита. (Этап классической коррекции ошибок обеспечивает защиту от ошибок измерения. Например, если блок находится в состоянии $|0\rangle_{\text{code}}$, то при измерении элементарных кубитов для определения логического кубита могут возникнуть две независимые ошибки, в результате чего будет получено неверное значение $|1\rangle_{\text{code}}$.)

В применении к квантовым вычислениям нам потребуется выполнять измерение, проецирующее на базис $\{|0\rangle_{\text{code}}, |1\rangle_{\text{code}}\}$ без разрушения блока. Эта выполняется путем копирования четности блока на служебный кубит с последующим его измерением. Схема, представляющая неразрушающее измерение кодового блока [в случае, когда матрица проверки четности аналогична описанной в уравнении (15)], показана на рисунке 4. Измерение является неразрушающим в том смысле, что оно сохраняет кодовое подпространство; конечно, оно «разрушает» когерентную суперпозицию $a|0\rangle_{\text{code}} + b|1\rangle_{\text{code}}$, коллапсируя это состояние либо к $|0\rangle_{\text{code}}$ (с вероятностью $|a|^2$), либо к $|1\rangle_{\text{code}}$ (с вероятностью $|b|^2$).

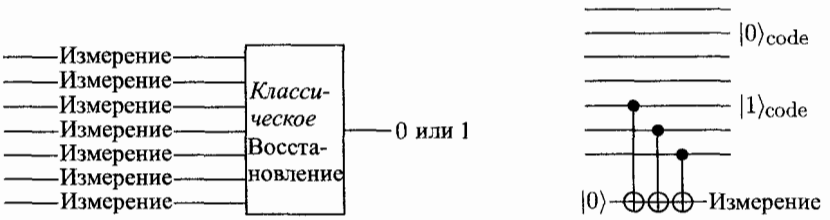


Рис. 4. Разрушающее и неразрушающее измерения логического кубита

7-кубитовый код Стина может исправить только одну ошибку в кодовом блоке, но можно построить лучшие коды [12, 28–31], способные защитить информацию от t ошибок в одном блоке, так что закодированная информация может сохраняться с точностью воспроизведения $F = 1 - O(\epsilon^{t+1})$.

Ключевая идея, сделавшая возможной коррекцию квантовых ошибок, состоит в том, что *с запутыванием можно бороться с помощью запутывания*. Запутывание может быть нашим врагом, поскольку запутывание устройства с окружением может скрыть от нас квантовую информацию и таким образом послужить причиной появления ошибок. Но запутывание может оказаться и нашим союзником — информацию, которую мы хотим защитить, можно закодировать в запутанном состоянии, то есть в корреляциях, охватывающих большое число кубитов. Тогда она остается недоступной при измерении лишь нескольких кубитов. По тем же причинам эта информация не может быть *повреждена*, если окружение взаимодействует лишь с несколькими кубитами.

Более того, мы узнали, что хотя квантовый компьютер в определенном смысле является аналоговым устройством, совершаемые им ошибки можно *оцифровать*. Мы боремся с малыми ошибками, выполняя подходящие измерения, проецирующие состояние квантового компьютера либо на исходное неповрежденное, либо на состояние с большой ошибкой, которую затем можно исправить известными методами. Мы также увидели, что можно измерять ошибки, не измеряя при этом данные, — можно получить точную информацию о природе ошибки, ничего не узнав о закодированной в нашем устройстве квантовой информации (что могло бы повлечь за собой декогерентизацию и сбой в вычислении).

Все квантовые коды коррекции ошибок используют одну и ту же фундаментальную стратегию: маленькое подпространство гильбертова пространства устройства определяется в качестве кодового подпространства. Это пространство тщательно выбирается, так чтобы все ошибки, которые

мы хотим исправить, перемещали его во взаимно ортогональные *подпространства ошибок*. После того, как наша система провзаимодействовала с окружающей средой, можно выполнить измерение, которое сообщит, в каком из этих взаимно ортогональных подпространств оказалась система и, следовательно, однозначно определить тип возникшей ошибки. После этого ошибку можно исправить, применив подходящее унитарное преобразование.

3. Отказоустойчивое исправление

До сих пор в наших обсуждениях предполагалось, что мы можем безошибочно кодировать квантовую информацию и выполнять исправление ошибок. Но, конечно, эти процедуры не будут выполняться безупречно. Восстановление само по себе является предрасположенным к появлению ошибок квантовым вычислением. Если вероятность появления ошибки для каждого бита в кодовом блоке равна ϵ, то естественно предположить, что каждый применяемый в ходе восстановления квантовый вентиль с вероятностью ϵ вызывает появление ошибки (или, что «ошибки запоминающего устройства» возникают в ходе исправления с вероятностью порядка ϵ). Если процедура исправления сконструирована небрежно, то вероятность ее отказа (например, вследствие возникновения двух ошибок в одном блоке) может иметь порядок ϵ. Тогда мы не извлечем пользы от применения квантовых кодов коррекции ошибок; в действительности вероятность появления ошибки на один кубит информации будет даже выше, чем при отсутствии какого-либо кодирования. Поэтому мы должны проанализировать все возможные причины, по которым процедуры исправления могут дать сбой с вероятностью порядка ϵ, и убедиться, что они устранены. Только тогда они будут *отказоустойчивыми*, и только тогда кодирование окупит себя при достаточно малой величине ϵ.

3.1. Проблема обратного действия

Распространение ошибок является серьезной проблемой. Если в одном кубите возникает ошибка, а затем мы применяем вентиль, в котором этот кубит взаимодействует с другим, то существует конечная вероятность распространения ошибки на второй кубит. Следует соблюдать осторожность, чтобы сдержать инфекцию, или, по крайней мере, нужно стремиться предотвратить возникновение двух ошибок в одном блоке.

При выполнении исправления ошибок мы многократно используем двухкубитовый XOR-вентиль. Этот вентиль способен распространять ошибки в двух различных направлениях. Во-первых, очевидно, что если

в одном кубите возникает ошибка инвертирования бита, а затем он используется в качестве источника XOR-вентиля, то инвертирование бита перейдет «в прямом направлении» на кубит цели. Более тонким является второй тип распространения ошибок; его можно понять, используя тождество, представленное на рисунке 5: при повороте Адамара базисов обоих кубитов источник и цель XOR-вентиля меняются ролями. Поскольку мы помним, что эта замена базиса также заменяет ошибку инвертирования бита на фазовую ошибку, мы делаем вывод, что при возникновении в одном кубите фазовой ошибки и последующем использовании его в качестве целевого кубита XOR-вентиля ошибка переходит на кубит источника.

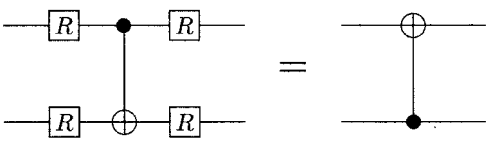


Рис. 5. Полезное тождество. Источник и цель XOR-вентиля меняются местами, если мы с помощью поворота Адамара выполняем замену базисов

Рис. 6. Плохой и хороший варианты измерения синдрома. Плохая схема использует один и тот же служебный бит несколько раз; хорошая схема использует служебный бит лишь однократно

Сейчас мы можем понять, что изображенная на рисунке 2 схема не является отказоустойчивой. Проблема состоит в том, что один и тот же служебный кубит используется в качестве цели четырех последовательных XOR-вентилей. Если на определенной стадии в служебном кубите возникает хотя бы одна фазовая ошибка, эта единственная ошибка может вернуть-

ся обратно к двум или более кубитам кодового блока данных. В результате с вероятностью ϵ в блоке может возникнуть фазовая ошибка, что недопустимо.

Чтобы понизить вероятность отказа до величины порядка ϵ^2, мы должны так изменить схему исправления, чтобы каждый служебный кубит взаимодействовал не более чем с одним кубитом в блоке. Один из способов решения этой задачи состоит в увеличении количества служебных кубитов от одного до четырех, чтобы каждый из них являлся целью одного XOR-вентиля, как это показано на рисунке 6. Затем мы можем измерить все четыре служебных кубита. Бит искомого нами синдрома является битом *четности* четырех измеряемых кубитов. По сути, мы скопировали из блока данных в служебный кубит некоторую информацию о возникшей ошибке, а при измерении служебного кубита мы считываем эту информацию.

Тем не менее, эта процедура по-прежнему неадекватна поставленной цели, поскольку мы скопировали *слишком много* информации. Схема запутывает служебный кубит с возникшей в данных ошибкой, что хорошо, но она запутывает его и с самими данными, а это плохо. Измерение служебного кубита разрушает тщательно приготовленную суперпозицию базисных состояний (6) и (7) для $|0\rangle_{\text{code}}$ и $|1\rangle_{\text{code}}$. Пусть, например, мы измеряем первый бит синдрома, как это показано на рисунке 2, но с увеличенным от одного до четырех количеством служебных кубитов. Следовательно, на самом деле мы измеряем последние четыре бита в блоке. Если в результате измерения мы получаем, скажем, $|0000\rangle_{\text{anc}}$, то это означает, что мы спроецировали $|0\rangle_{\text{code}}$ на $|0000000\rangle$, а $|1\rangle_{\text{code}}$ на $|1110000\rangle$; кодовые слова теряют всякую защиту от фазовых ошибок.

3.2. Приготовление служебного состояния

Продолжим модификацию процедуры исправления, сохраняя ее хорошие черты и исключая плохие. Мы хотим скопировать на служебные кубиты информацию об ошибках, возникающих в блоке данных, не внося в него многочисленных фазовых ошибок и не разрушая когерентности закодированных в нем данных. Для достижения этой цели необходимо, прежде чем приступать к вычислению синдрома ошибки, приготовить подходящее состояние служебных кубитов. Это состояние строится таким образом, чтобы результат его измерения открывал информацию об ошибках, ничего не сообщая о состоянии данных.

Один из способов удовлетворить этому критерию был найден Питером Шором. Он предложил использовать в качестве состояний четырех служебных кубитов равновзвешенную суперпозицию всех кодовых слов четного

веса (*состояние Шора*)

$$|\mathrm{Shor}\rangle_{\mathrm{anc}} = \frac{1}{\sqrt{8}} \sum_{v_{\mathrm{even}}} |v_{\mathrm{even}}\rangle_{\mathrm{anc}}. \tag{16}$$

Чтобы вычислить каждый бит синдрома, мы готовим служебные кубиты в состояниях Шора, выполняем четыре XOR-вентиля (с подходящими кубитами блока данных в качестве источников и четырьмя кубитами в состояниях Шора в качестве целей), а затем измеряем служебное состояние.

Если вычисляемый бит тривиален, то к кодовым словам v в (16) добавляется строка четного веса, что оставляет состояние Шора неизменным; если бит синдрома нетривиален, тогда состояние Шора преобразуется в равновзвешенную суперпозицию кодовых слов с нечетным весом. Таким образом, четность результата измерения выявляет значение бита синдрома, но никакую другую информацию о состоянии блока данных из этого измерения извлечь нельзя — мы нашли способ выделить синдром, не повреждая кодовые слова. (Определяемая нами при измерении отдельная строка с заданной четностью выбирается случайным образом и не имеет ничего общего с состоянием блока данных.)

Всего существует шесть битов синдрома (по три для диагностики ошибок инвертирования бита и для диагностики ошибок обращения фазы), поэтому измерение синдрома использует 24 служебных бита, приготовленных в шести состояниях Шора, и 24 XOR-вентиля.

Одним из способов получения синдрома обращения фазы может быть следующий: сначала применить к блоку данных семь параллельных R-вентилей для поворота базиса, затем, как показано на рисунке 2, применить XOR-вентили (но с приготовленными в состояниях Шора служебными кубитами) и, наконец, применить семь R-вентилей для поворота данных в исходный базис. Однако с целью усовершенствования этой процедуры мы можем использовать представленное на рисунке 5 тождество. Обращая направление XOR-вентилей (то есть используя служебные биты в качестве источников, а данные — в качестве цели), мы можем избежать применения R-вентилей к данным и, следовательно, понизить вероятность их повреждения [16, 24], как это показано на рисунке 7.

Другой способ приготовления служебного кубита был предложен Эндрю Стином. Его 7-кубитовое служебное состояние является равновзвешенной суперпозицией кодовых слов Хэмминга:

$$|\mathrm{Steane}\rangle_{\mathrm{anc}} = \frac{1}{4} \sum_{v \in \mathrm{H}} |v\rangle. \tag{17}$$

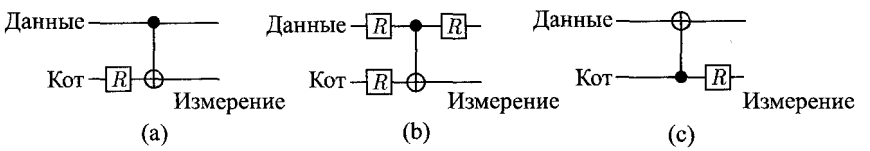


Рис. 7. (a) Схематическое изображение процедуры вычисления одного бита синдрома ошибки инвертирования бита. Применяемый к «кот-состоянию» вентиль Адамара завершает приготовление состояния Шора (см. раздел 3.3). Как XOR-вентиль, так и вентиль Адамара на диаграмме фактически представляют собой четыре параллельно выполняемых вентиля. (b) Схематическое изображение процедуры вычисления одного бита синдрома ошибки обращения фазы. Она аналогична (a), но применяется к данным в базисе, повернутом преобразованием Адамара. (c) Эквивалентная (b) схема, упрощенная с помощью тождества, изображенного на рисунке 5

(Это состояние можно также записать в виде $(|0\rangle_{\text{code}} + |1\rangle_{\text{code}})/\sqrt{2}$; его можно получить, применив к состоянию $|0\rangle_{\text{code}}$ побитовый поворот Адамара.) Чтобы вычислить синдром инвертирования бита, мы с помощью XOR-вентиля обращаем каждый кубит блока данных в соответствующий кубит служебного состояния, а затем измеряем его. Применяя к результату *классического* измерения матрицу проверки четности Хэмминга H, мы выделяем синдром инвертирования бита. Как и в методе Шора, эта процедура «копирует» данные в служебный кубит, состояние которого выбрано таким образом, чтобы при его измерении обеспечить прочтение информации только об ошибке. Например, если ошибка отсутствует, найденная в измерении конкретная строка является случайно выбранным кодовым словом Хэмминга и ничего не сообщает о состоянии данных. При определении синдрома обращения фазы аналогичная процедура выполняется в повернутом базисе. Преимущество метода Стина перед процедурой Шора состоит в том, что он требует лишь 14 служебных битов и 14 XOR-вентилей. Но он также имеет и недостаток: приготовление состояния Стина сложнее, чем приготовление состояния Шора, так что оно отчасти более подвержено возникновению ошибок.

3.3. Проверка служебного состояния

Продолжая изучать все ситуации, в которых сбой процесса восстановления может произойти из-за единственной ошибки, мы обнаруживаем другую потенциальную проблему. Вследствие распространения ошибок, единственная ошибка, возникающая в процессе приготовления состояния Шора

или состояния Стина, может послужить причиной появления в них двух фазовых ошибок, а они, в свою очередь, могут распространиться на закодированные данные, если для измерения синдрома будет использован дефектный служебный кубит. Наша процедура по-прежнему не является отказоустойчивой.

Следовательно, перед его использованием состояние служебных кубитов должно быть протестировано на многократные фазовые ошибки. Если оно не выдерживает тест, оно должно быть забраковано, а служебное состояние построено заново.

Один из способов приготовления и проверки состояния Шора изображен на рисунке 8. Первый вентиль Адамара и первые три XOR-вентиля в этой схеме готовят «кот-состояние» ($|0000\rangle + |1111\rangle$), максимально запутанное состояние четырех служебных кубитов; последние четыре вентиля Адамара преобразуют кот-состояние в состояние Шора. Но единственная ошибка, возникающая в ходе выполнения второго или третьего XOR-вентиля, может повлечь за собой появление двух ошибок в кот-состоянии (оно может принять вид ($|0011\rangle + |1100\rangle$)). Эти две ошибки инвертирования бита в кот-состоянии превращаются в две фазовые ошибки в состоянии Шора, что по возвращении в исходное состояние во время измерения синдрома приведет к фазовой ошибке в блоке.

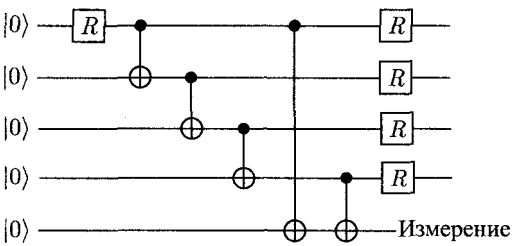


Рис. 8. Приготовление и проверка состояния Шора. Если результат измерения равен 1, тогда это состояние бракуется и готовится новое

Заметим, однако, что во всех случаях, когда один плохой вентиль может вызвать в кот-состоянии две ошибки инвертирования бита, разные значения будут иметь первый и четвертый биты кот-состояния. Следовательно, мы добавляем к схеме два последних XOR-вентиля (сопровождаемые измерением), чтобы проверить соответствие этих двух битов кот-состояния. Если проверка успешна, мы можем перейти к измерению синдрома, уверенные в том, что вероятность возникновения двух фазовых ошибок в состоя-

нии Шора имеет порядок ϵ^2. Если проверка дает сбой, мы можем выкинуть кот-состояние и совершить новую попытку.

Конечно, единственная ошибка в схеме приготовления также может привести к появления двух *фазовых* ошибок в кот-состоянии и, следовательно, двух ошибок инвертирования бита в состоянии Шора; мы не пытались проверять состояние Шора на наличие ошибок инвертирования битов. Но, по сравнению с фазовыми ошибками, они нас беспокоят гораздо меньше. Ошибки инвертирования бита становятся причиной ошибочного измерения синдрома, но они не распространяются в обратном направлении и не повреждают закодированную информацию.

Если для измерения синдрома используется метод Стина, то сначала с помощью изображенной на рисунке 3 кодирующей схемы (но без первых двух XOR-вентилей) строится состояние $|0\rangle_{\text{code}}$, а затем, чтобы завершить приготовление состояния Стина, к каждому кубиту $|0\rangle_{\text{code}}$ применяется вентиль Адамара. Поскольку вновь единственная ошибка, возникающая в процессе кодирования, может привести к появлению в $|0\rangle_{\text{code}}$ двух ошибок инвертирования бита, которые в состоянии Стина преобразуются в две фазовые ошибки, то на этом этапе также необходима проверка. Ее можно осуществить, выполняя неразрушающее измерение, которое гарантирует, что (с точностью до одного инвертирования бита) приготовленным состоянием является $|0\rangle_{\text{code}}$, а не $|1\rangle_{\text{code}}$. С этой целью готовятся два блока в состоянии $|0\rangle_{\text{code}}$, затем производится побитовое выполнение XOR-вентиля с первым блоком в качестве источника и вторым — в качестве цели и, наконец, выполняется разрушающее измерение второго блока. Применяя к результатам этого измерения классический корректирующий код Хэмминга, можно исправить одну возможную ошибку инвертирования бита и идентифицировать измеренный блок как $|0\rangle_{\text{code}}$ или $|1\rangle_{\text{code}}$. Если результатом последнего измерения является $|0\rangle_{\text{code}}$, то это означает, что другой блок прошел проверку. В противном случае считается, что он поврежден и должен быть восстановлен путем инвертирования битов.

Однако эта процедура проверки все еще ненадежна, поскольку на самом деле поврежденным может оказаться измеряемый нами блок, а не тот, который мы пытаемся проверить. Следовательно, необходимо повторить этап проверки. Если измеряемый блок выдает один и тот же результат дважды подряд, то проверка может считаться надежной. А что, если во второй раз будет получен другой результат? Тогда непонятно, что делать, инвертировать биты проверяемого блока или оставить его в покое. Можно сделать еще одну попытку, чтобы разорвать этот узел, но на самом деле в этом нет необходимости; в действительности, если две попытки исправления дают противоречивые результаты, то лучше ничего не предпринимать. Поскольку

результаты противоречат друг другу, то понятно, что один из двух измеряемых блоков поврежден. Следовательно, вероятность того, что проверяемый блок тоже ошибочный, имеет порядок ϵ^2, и ею можно пренебречь. Итак, с помощью этой процедуры проверки удается построить такое состояние Стина, в котором вероятность появления многократных фазовых ошибок (способных распространяться на закодированные данные при измерении синдрома) имеет порядок ϵ^2.

3.4. Проверка синдрома

Одна ошибка инвертирования бита в служебном состоянии может привести к повреждению синдрома. Ошибка может возникнуть из-за неправильного приготовления служебного состояния или вследствие ошибки, появившейся в процессе вычисления синдрома. Последний случай особенно опасен, потому что возникающая с вероятностью порядка ϵ единственная ошибка может повредить как блок данных, так и служебное состояние. Это может случиться, например, вследствие того, что плохой XOR-вентиль одновременно вносит ошибки в источник и цель, или из-за того, что в блоке данных в ходе измерения синдрома возникает ошибка, которая затем распространяется XOR-вентилем на состояние служебных кубитов.

В таком случае, если бы мы воспользовались поврежденным синдромом для исправления ошибки, то на самом деле ввели бы в блок закодированных данных дополнительную ошибку. Поэтому наша процедура остается не полностью отказоустойчивой; возникающая с вероятностью порядка ϵ ошибка может привести к фатальному повреждению данных.

Следовательно, необходимо найти возможность обеспечить более надежный синдром. Очевидным способом является повторение измерения синдрома. Оно не обязательно, если результат измерения синдрома тривиален (не сообщает об ошибке); даже если в блоке данных имеется ошибка, которая оказалась не замеченной, не стоит беспокоиться, что это усугубит ситуацию, просто потому, что в этом случае не совершается никаких действий. С другой стороны, если синдром сообщает об ошибке, то его измерение повторяется. В этом случае, если вновь получается тот же самый результат, можно с уверенностью принимать синдром и переходить к исправлению, поскольку невозможно получение с вероятностью ϵ одного и того же (нетривиального) ошибочного синдрома.

Если первые два измерения синдрома не согласуются, тогда можно продолжать его измерение до тех пор, пока один и тот же результат не будет получен два раза подряд — результат, которому можно доверять. В качестве альтернативы, можно ничего не предпринимать, пока ошибка не

будет надежно детектирована в следующем раунде коррекции. По крайней мере, это бездействие не усугубит ситуацию, а ошибка, если она на самом деле присутствует в закодированных данных, возможно, будет обнаружена в следующий раз.

Существуют и другие способы повысить нашу уверенность в синдроме. Например, вместо повторения измерения целого синдрома можно было бы вычислить некоторые дополнительные избыточные биты синдрома и подвергнуть их проверке на четность. Если в синдроме содержится ошибка, то этот метод, как правило, ее обнаружит; таким образом, если проверка на четность проходит успешно, то корректность синдрома является достаточно вероятной [24, 32].

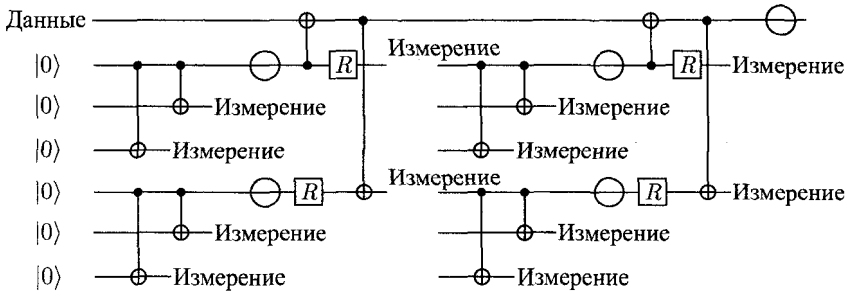


Рис. 9. Полная схема исправления ошибок согласно Стину. Готовятся и затем проверяются закодированные состояния $|0\rangle$. Проверенные $|0\rangle$ используются в качестве служебных кубитов для вычисления синдромов как инвертирования битов, так и обращения фазы каждый; измеряется дважды. Большие круги обозначают действия (обусловленные результатами измерения), предпринимаемые для восстановления служебных состояний, или действия по восстановлению блока данных на финальном этапе

Итак, мы собрали вместе все элементы отказоустойчивой процедуры исправления ошибок. Если мы предпримем все описанные выше меры предосторожности, то исправление даст сбой только при появлении двух независимых ошибок, поэтому вероятность необратимого повреждения закодированного блока имеет порядок ϵ^2.

Полная квантовая схема исправления ошибок согласно Стину показана на рисунке 9. Отметим, что исправление ошибок как инвертирования бита, так и обращения фазы повторяются дважды. На схеме показана и проверка состояний Стина, но этап кодирования этих состояний опущен.

3.5. Измерение и кодирование

Конечно, мы хотим уметь надежно измерять наши закодированные кубиты. Но как уже отмечалось в разделе 2, *разрушающее* измерение кодового блока надежно, если только один кубит в блоке имеет ошибку инвертирования бита. Если в случае одного кубита вероятность ошибочного измерения имеет порядок ϵ, то ошибочные измерения кодового блока возникают с вероятностью порядка ϵ^2. Отказоустойчивое неразрушающее измерение также может быть выполнено, как уже отмечалось при обсуждении проверки состояния Стина (раздел 3.3). В качестве альтернативной процедуры можно было бы без каких-либо модификаций использовать изображенное на рис. 4 неразрушающее измерение. Хотя служебный бит является целью трех последовательных XOR-вентилей, возвращающиеся в блок фазовые ошибки не так губительны, поскольку они не могут заменить $|0\rangle_{\text{code}}$ на $|1\rangle_{\text{code}}$ (или наоборот). Но, поскольку единственная ошибка инвертирования бита (как в блоке данных, так и в служебном кубите) может вызвать ошибку измерения четности, для обеспечения точности порядка ϵ^2 измерение необходимо повторить (после исправления ошибки инвертирования бита). (Мы умолчали об этой процедуре в описании проверки состояния Стина, чтобы избежать разочарования от необходимости коррекции ошибок при подготовке служебных кубитов, предназначенных для коррекции ошибок!)

Часто нам будет необходимо приготовить известное закодированное квантовое состояние, например, $|0\rangle_{\text{code}}$. Мы только что обсудили (в разделе 3.3 в связи с приготовлением состояния Стина), как можно надежно выполнить это кодирование. В сущности, кодирующая схема не нужна. Каким бы ни было исходное состояние блока, (отказоустойчивое) исправление ошибок спроецирует его на натянутое на $\{|0\rangle_{\text{code}}, |1\rangle_{\text{code}}\}$ пространство, а (проверенное) измерение выдаст либо $|0\rangle_{\text{code}}$, либо $|1\rangle_{\text{code}}$. Если получен результат $|1\rangle_{\text{code}}$, то для обращения блока в желаемое состояние $|0\rangle_{\text{code}}$ можно применить (побитовый) NOT-оператор.

Если мы хотим закодировать неизвестное квантовое состояние, то мы используем кодирующую схему, изображенную на рисунке 3. Снова, вследствие распространения ошибок, единственная ошибка в процессе кодирования может привести к сбою. В данном случае, поскольку никакое измерение не может проверить кодирование, точность воспроизведения закодированного состояния будет неизбежно равна $F = 1 - O(\epsilon)$. Тем не менее, смысл в кодировании не теряется, поскольку оно позволяет хранить состояние с достаточной точностью воспроизведения в течение более продолжительного времени, чем если бы состояние оставалось незакодированным.

3.6. Другие коды

Схемы Шора и Стина для отказоустойчивого измерения синдрома выше были описаны лишь для 7-кубитового кода, но их можно адаптировать и для более сложных кодов, способных исправлять множество ошибок [16, 33]. Измерение синдрома для более общих кодов лучше всего описывается с использованием формализма стабилизаторов. Согласно этому формализму, каждый генератор можно представить в виде произведения действующих на отдельный кубит операторов, где однокубитовые операторы выбраны из определяемого уравнением (5) множества $\{I, X, Y, Z\}$. Каждый генератор при возведении в квадрат дает единицу и имеет одинаковые количества собственных векторов с собственными значениями $+1$ и -1, так что определение его собственного значения вдвое сокращает размерность пространства. Если в блоке содержится n кубитов и имеется $n-k$ генераторов, то кодовое подпространство имеет размерность 2^k, то есть имеется k закодированных кубитов.

Например, 7-кубитовый код Стина представляет собой пространство, в котором все шесть генераторов стабилизатора

$$\begin{aligned} M_1 &= (IIIZZZZ), \\ M_2 &= (IZZIIZZ), \\ M_3 &= (ZIZIZIZ), \\ M_4 &= (IIIXXXX), \\ M_5 &= (IXXIIXX), \\ M_6 &= (XIXIXIX) \end{aligned} \tag{18}$$

имеют собственное значение $+1$. Сравнение с (1) показывает, что пространство с $M_1 = M_2 = M_3 = 1$ натянуто на кодовые слова, удовлетворяющие проверке четности Хэмминга, а поскольку адамаровский поворот базиса меняет местами Z и X, то пространство с $M_4 = M_5 = M_6 = 1$ натянуто на кодовые слова, удовлетворяющие проверке четности Хэмминга в повернутом базисе. Действительно, определяющим свойством кода Стина является то, что проверка четности Хэмминга удовлетворяется в обоих случаях.

Генераторы стабилизатора выбираются таким образом, чтобы оператор каждой подлежащей исправлению ошибки (также представляемый в виде произведения однокубитовых операторов $\{I, X, Y, Z\}$), а также произведение любых двух различных операторов ошибок антикоммутировали по крайней мере с одним генератором. Таким образом, каждая ошибка изменяет собственное значение одного из генераторов, а две независимые ошибки

всегда изменяют собственные значения своим, отличным от других, способом. Это означает, что измерение собственных значений всех генераторов стабилизатора дает полный синдром ошибки.[1]

Измерение генератора стабилизатора M не сложно. Сначала выполняется подходящее унитарное преобразование каждого кубита, так чтобы в новом базисе оператор M представлял собой произведение действующих на отдельные кубиты I и Z. (Для каждого кубита, находящегося под действием X в M, мы совершаем поворот

$$R = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{19}$$

а также поворот

$$R' = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \tag{20}$$

для каждого кубита, находящегося под действием Y.) В этом базисе собственное значение M представляет собой четность битов, на которые действуют Z. Ее можно измерить (примерно так же, как это делалось при обсуждении 7-кубитового кода), применяя к служебным кубитам XOR-вентили, использующие в качестве источников кубиты в блоке, на которые в новом базисе действуют операторы Z из M. После возвращения в исходный базис эта процедура повторяется для следующего генератора стабилизатора, и так далее до тех пор, пока не будет получен полный синдром.

Мы можем сделать эту процедуру отказоустойчивой, заготовив для измерения каждого бита синдрома служебное состояние Шора, количество кубитов в котором равно *весу* соответствующего генератора стабилизатора (количеству не равных единичному однокубитовых операторов). Каждый служебный бит является целью лишь одного XOR-вентиля, так что многократные фазовые ошибки не возвращаются в данные. Рассмотренные выше процедуры проверки состояния Шора и измерения синдрома также можно соответствующим образом обобщить.

Для сложных кодов, которые либо кодируют множество кубитов, либо исправляют большое количество ошибок, обобщенный метод Шора требует значительно больше служебных кубитов и квантовых вентилей, чем

[1] На самом деле произведение операторов двух независимых ошибок может принадлежать стабилизатору. Тогда эти две ошибки будут иметь один и тот же синдром. Однако это не имеет значения, поскольку они также могут быть исправлены одним и тем же действием. Квантовые коды, сопоставляющие один синдром нескольким различным операторам ошибок, называются вырожденными.

это необходимо для выявления синдрома ошибки. В этом случае гораздо лучше воспользоваться обобщенным методом Стина. Идея Стина состоит в том, что в случае 7-кубитового кода для измерения всех M_1, M_2 и M_3 можно использовать одно 7-кубитовое служебное состояние. Действительно, приготовим исходное служебное состояние в виде равновзвешенной суперпозиции всех строк, удовлетворяющих проверке четности Хэмминга (то есть всех слов классического кода Хэмминга) применим к нему подходящие XOR-вентили, использующие кубиты блока в качестве источников, измерим все кубиты служебного состояния и, наконец, применим проверку четности Хэмминга к результату измерения. Три полученных бита четности представляют собой собственные значения операторов M_1, M_2 и M_3. Служебное состояние здесь приготовлено таким образом, чтобы, помимо этих собственных значений, никакая другая информация не могла быть извлечена из результата измерения; следовательно, эта процедура не разрушает когерентность квантовых кодовых слов.

Очевидно, эту процедуру можно адаптировать для одновременного измерения собственных значений любой совокупности операторов, имеющих вид произведений I и Z, действующих на отдельные кубиты. С этой целью по заданному списку k таких n-кубитовых операторов M_i $(1 \leqslant i \leqslant k)$ построим $k \times n$-матрицу H_Z, ij-элемент которой равен нулю (единице), если в j-й позиции M_i находится оператор I (Z). Затем приготовим n-кубитовое служебное состояние, представляющее собой равновзвешенную суперпозицию всех строк длины n, прошедших проверку четности матрицей H_Z. Применяя к этому состоянию соответствующие XOR-вентили и измеряя результат (а также применяя H_Z к результату измерения), спроецируем n-кубитовый блок на общее собственное состояние рассматриваемого семейства операторов M_i. Выполняя аналогичную процедуру в базисе, повернутом преобразованием Адамара, можно одновременно измерить собственные значения любой совокупности операторов, имеющих вид произведений I и X.

Среди генераторов стабилизатора могут быть и операторы, имеющие вид $M = \bar{Z}\bar{X}$, где $\bar{Z}$ — произведение операторов Z, действующих на одну совокупность кубитов, а $\bar{X}$ — произведение операторов X, действующих на другую совокупность кубитов. Поскольку квадрат каждого генератора M должен быть равен единице, число кубитов, на которые действует оператор Y, равный произведению Z и X, должно быть четным. Следовательно, $\bar{Z}$ и $\bar{X}$ коммутируют, и их можно измерить при помощи описанного выше метода. Однако такое измерение даст слишком много информации; мы хотим измерить произведение $\bar{Z}\bar{X}$, а не каждый из сомножителей $\bar{Z}$ и $\bar{X}$ в отдельности. Для осуществления желаемого измерения необходима дальнейшая модификация служебного состояния. Оно не должно выби-

раться одновременно удовлетворяющим проверке четности матрицами H_Z и H_X. Скорее, его следует приготовить так, чтобы биты четности H_Z и H_X были коррелированы, то есть служебное состояние должно представляться суммой таких строк, у которых оба бита четности либо тривиальны, либо нетривиальны. После измерения служебного состояния собственное значение оператора M получается суммированием четностей «$\bar{Z}$- и $\bar{X}$-измерения». Индивидуальные четности $\bar{Z}$- и $\bar{X}$- измерений являются абсолютно случайными и фактически ничего не сообщают о значениях операторов $\bar{Z}$ и $\bar{X}$.

Теперь мы можем описать метод Стина в его общей форме, в которой он применим к любому стабилизирующему коду. Если k логических кубитов закодированы в блоке из n кубитов, то существует $n-k$ независимых генераторов стабилизатора. Со списком этих генераторов ассоциируется матрица

$$\bar{H} = (H_Z \,|\, H_X), \tag{21}$$

имеющая $n-k$ строк и $2n$ столбцов. Позиции единиц в H_Z обозначают кубиты, на которые в перечисленных генераторах действует Z, а положения единиц в H_X обозначают кубиты, находящиеся под действием X; если единица возникает в одном и том же положении в H_Z и H_X, то на этот кубит действует произведение $Y = ZX$. Далее, готовится служебное $2n$-кубитовое *обобщенное состояние Стина* — равновзвешенная суперпозиция всех строк, удовлетворяющих проверке четности $\bar{H}$. Затем выполняется изображенная на рисунке 10 квантовая схема, измеряются служебные кубиты и к результату измерения применяется $\bar{H}$. Полученные биты четности являются собственными значениями генераторов стабилизатора, что дает полный синдром ошибки.

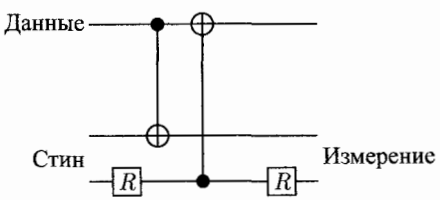


Рис. 10. Схема Стина для измерения синдрома. $2n$-кубитовое состояние Стина используется для определения синдрома для n-кубитового блока данных. Каждый XOR-вентиль на схеме представляет n выполняемых параллельно XOR-вентилей

Служебное состояние здесь приготовлено таким образом, что из результата измерения не может быть извлечена никакая другая информация,

кроме синдрома, и, следовательно, эта процедура не нарушает когерентность квантовых кодовых слов. В этой процедуре на каждый кубит в кодовом блоке действуют только два квантовых вентиля — необходимый минимум для определения как ошибки инвертирования бита, так и ошибки обращения фазы, поражающих любой кубит.

В заключение отметим, что Китаевым [34] была описана другая стратегия выполнения отказоустойчивого исправления ошибок. Он предложил семейство корректирующих ошибки квантовых кодов, способных исправить множество ошибок в кодовом блоке и требующих лишь четыре XOR-вентиля для вычисления каждого бита синдрома. В данном случае, даже если для вычисления каждого бита синдрома мы будем использовать лишь один служебный кубит (а не обобщенное служебное состояние, аналогичное состоянию Стина или Шора), из служебного кубита обратно в данные может вернуться лишь ограниченное количество ошибок. Тогда код можно выбрать таким образом, чтобы типичное количество ошибок, вернувшихся в данные в процессе вычисления синдрома, было заведомо меньше максимального числа ошибок, которое код еще может выдержать.

4. Отказоустойчивые квантовые вентили

Мы убедились, что кодирование может защитить квантовую информацию. Но мы хотим больше, нежели просто *хранить* квантовую информацию с высокой точностью воспроизведения; мы хотим оперировать квантовым компьютером, который *обрабатывает* эту информацию. Конечно, мы могли бы декодировать информацию, применить вентиль, а затем снова закодировать ее, но эта процедура на время подвергла бы опасности квантовую информацию. Если мы хотим, чтобы наш квантовый компьютер работал надежно, вместо этого,мы должны научиться применять квантовые вентили непосредственно к закодированным данным. В свою очередь, эти вентили должны удовлетворять принципам отказоустойчивости, если мы хотим избежать катастрофического распространения ошибок.

4.1. 7-кубитовый код

Действительно, существует ряд вентилей, которые можно легко применять с 7-кубитовым кодом Стина. Все три однокубитовых вентиля могут применяться *побитово*; то есть применение этих вентилей к каждому из семи кубитов в блоке обеспечивает применение того же вентиля к закодированному кубиту. Мы уже видели, что именно так действует поворот Адамара R [см. уравнение (11)]. То же самое можно сказать о вентиле NOT

(так как каждое нечетное кодовое слово Хэмминга является дополнением четного),[1] и вентиле сдвига фазы

$$P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{22}$$

(Нечетные кодовые слова Хэмминга имеют вес $\equiv 3 \pmod 4$, а четные — вес $\equiv 0 \pmod 4$, поэтому для выполнения P мы фактически применяем P^{-1} побитово.) XOR-вентиль также можно выполнять побитово, то есть путем применения к каждому кубиту блока целей своего XOR-вентиля, использующего соответствующий кубит из блока источников, как это показано на рисунке 11. Эта процедура работает, потому что четные кодовые слова образуют субкод, тогда как нечетные кодовые слова — его нетривиальный смежный класс.

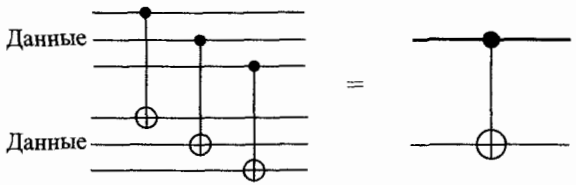


Рис. 11. Схематическое изображение трансверсального XOR-вентиля. Применяя XOR-вентиль к каждому кубиту из блока цели и соответствующему кубиту блока источника, мы выполняем XOR-вентиль, действующий на закодированные кубиты. Реализация вентиля отказоустойчива, поскольку на каждый кубит в обоих кодовых блоках действует свой вентиль

Таким образом, существуют простые отказоустойчивые процедуры выполнения вентилей NOT, R, P и XOR. Но к сожалению, сами по себе они не образуют универсального набора. Для того чтобы иметь возможность выполнять произвольные унитарные преобразования закодированной квантовой информации (с произвольной точностью), этот набор необходимо соответствующим образом дополнить. Следуя Шору [15], добавим к нему 3-кубитовый вентиль Тоффоли, который выполняется с помощью изображенной на рисунке 13 процедуры.[2]

Конструкция Шора отказоустойчивого вентиля Тоффоли предусматривает два этапа. На первом этапе три закодированных служебных блока го-

[1] Собственно, мы можем применить операцию NOT, действующую на закодированный кубит, при помощи лишь трех NOT, применяемых к выбранным кубитам в блоке.

[2] Нилл и др. [19, 20] описывают альтернативный способ комплектования универсального набора вентилей.

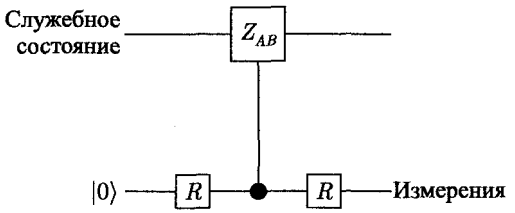


Рис. 12. Схема измерения, используемая на этапе приготовления служебного состояния для предложенной Шором реализации вентиля Тоффоли

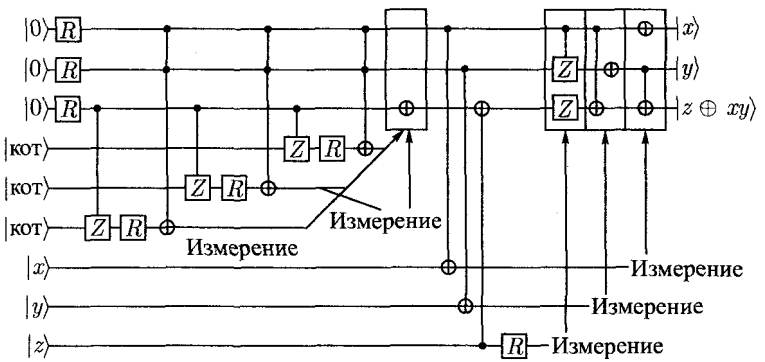


Рис. 13. Отказоустойчивый вентиль Тоффоли. Каждая линия представляет блок из семи кубитов, а вентили выполняются трансверсально. Для каждого измерения стрелка указывает на совокупность вентилей, которые применяются, если результатом измерения является 1, и не действуют, если результатом является 0

товятся в состоянии вида

$$|A\rangle_{\text{anc}} \equiv \frac{1}{2} \sum_{a=0,1} \sum_{b=0,1} |a, b, ab\rangle_{\text{anc}}. \tag{23}$$

На втором этапе, чтобы завершить выполнение вентиля, служебное состояние взаимодействует с тремя блоками данных. Прежде всего опишем способ приготовления служебного состояния. Для начала каждый из трех служебных блоков кодируется в состоянии $|0\rangle_{\text{code}}$. Для приготовления закодированного состояния ко всем трем блокам применяются побитовые вентили

Адамара

$$\frac{1}{\sqrt{8}} \sum_{a=0,1} \sum_{b=0,1} \sum_{c=0,1} |a,b,c\rangle_{\text{anc}}. \tag{24}$$

Отметим, что это состояние можно представить в виде

$$\frac{1}{\sqrt{2}}(|A\rangle_{\text{anc}} + |B\rangle_{\text{anc}}), \quad |B\rangle_{\text{anc}} \equiv \text{NOT}_3|A\rangle_{\text{anc}}, \tag{25}$$

где NOT_3 обозначает NOT-вентиль, действующий на третий закодированный кубит. В завершение приготовления служебного состояния эти три блока измеряются в базисе $\{|A\rangle_{\text{anc}}, |B\rangle_{\text{anc}}\}$; если получается результат $|A\rangle_{\text{anc}}$, приготовление завершено, если результат измерения $|B\rangle_{\text{anc}}$, для завершения процедуры применяется вентиль NOT_3.

Теперь объясним, как выполняется измерение в базисе $\{|A\rangle_{\text{anc}}, |B\rangle_{\text{anc}}\}$. Схематически оно осуществляется с помощью показанной на рисунке 12 схемы, где вентиль Z_{AB} (обусловленный управляющим битом) обращает относительную фазу $|A\rangle_{\text{anc}}$ и $|B\rangle_{\text{anc}}$. Из уравнений (23) и (25) можно увидеть, что Z_{AB} применяет фазовый множитель $(-1)^{ab+c}$, выражающийся через значения трех служебных блоков a, b и c. Если контрольный бит обозначается x, тогда вентилями, которые нам необходимо применить, являются $(-1)^{xab}$ и $(-1)^{xc}$ — произведение трехбитового и двухбитового фазовых вентилей.

Но трехбитовый фазовый вентиль так же сложен, как и вентиль Тоффоли, так что, похоже, мы попали в тупик. Однако из него можно выйти, выбрав управляющий блок таким образом, чтобы им был не закодированный кубит, а (проверенное) 7-битовое «кот-состояние»

$$|\text{cat}\rangle = \frac{1}{\sqrt{2}}(|0000000\rangle + |1111111\rangle). \tag{26}$$

Мы уже знаем, каким образом построить трансверсально действующие отказоустойчивые *двух-* и *однокубитовые* фазовые вентили. Их можно повысить до необходимых нам трех- и двухкубитовых вентилей простым управлением всех побитовых вентилей конструкции соответствующими битами кот-состояния. Наконец, мы применяем побитовый поворот Адамара к кот-состоянию и измеряем его четность, завершая таким образом выполнение схемы измерения на рисунке 12. (Мы получим схему на рисунке 13, заметив, что если кот-состояние находится в базисе полученном повернутом Адамара, то трехбитовый фазовый вентиль можно представить как вентиль Тоффоли с кот-состоянием в качестве цели; следовательно, при таком исполнении схемы измерения осуществляется один побитовый вентиль Тоф-

фоли). Конечно, для обеспечения точности необходимо повторение измерения.

Все это время три блока данных терпеливо ожидали готовности служебного кубита. Применением трех XOR-вентилей и поворота Адамара, состояние данных и служебного кубита преобразуются как

$$\begin{aligned}&\sum_{a=0,1}\sum_{b=0,1}|a,b,ab\rangle_{\text{anc}}|x,y,z\rangle_{\text{data}}\to\\&\to\sum_{a=0,1}\sum_{b=0,1}\sum_{w=0,1}(-1)^{wz}|a,b,ab\oplus z\rangle_{\text{anc}}|x\oplus a,y\oplus b,w\rangle_{\text{data}}.\end{aligned}\tag{27}$$

Теперь выполняется измерение каждого *блока данных*. Если результат измерения 0, то никакие действия не предпринимаются, но если результат измерения 1, то для завершения выполнения вентиля Тоффоли к служебному кубиту применяется определенный набор вентилей, как это показано на рисунке 13. Отметим, что эта процедура разрушает исходные блоки данных, а новыми блоками данных становятся те, что в начале являлись служебными. Важной особенностью этой конструкции является то, что все этапы тщательно построены в соответствии с принципами отказоустойчивости и минимизации распространения ошибок. Таким образом, для того чтобы в любом одном из блоков данных одновременно появились две ошибки, в ходе выполнения процедуры должны возникнуть две независимые ошибки.

Немного досадно, что отказоустойчивые вентили образуют дискретный набор, но это также неизбежная черта любой отказоустойчивой схемы. Нет смысла создавать континуум отказоустойчивых вентилей, поскольку как тогда мы сможем избежать ошибки, применяя неправильный вентиль, отличающийся от предполагаемого на малую величину? В любом случае, так как наши отказоустойчивые вентили образуют универсальный набор, их достаточно для приближения любого желаемого унитарного преобразования с любой желаемой точностью.

4.2. Другие коды

Шор [15] описал, как обобщить этот отказоустойчивый набор вентилей на более сложные коды, способные исправлять большее количество ошибок, а Готтесман [17,35] описал еще более общую процедуру, которую можно применить к любым квантовым стабилизирующим кодам.

Конструкция Готтесмана начинается со следующего наблюдения: для любого стабилизирующего кода существует отказоустойчивая реализация

однокубитовых вентилей X и Z, действующих на каждый закодированный кубит. Вспомним, что мы уже видели в разделе 3.6, что для стабилизирующего кода с размером блока n любой «оператор ошибки» M (представленный в виде тензорного произведения n матриц из набора $\{I, X, Y, Z\}$) можно записать в виде $\bar{Z}\bar{X}$ и, следовательно, однозначно представить в виде двоичной строки длины $2n$. Если имеется k закодированных в блоке логических кубитов, то стабилизатор кода генерируется $n-k$ такими операторами. Коммутирующие со всеми элементами стабилизатора операторы ошибок сами образуют группу. Генераторы этой группы представляются двоичными строками длины $2n$, которые должны удовлетворять $n-k$ независимым двоичным условиям; следовательно, имеется $n+k$ независимых генераторов. Из них $n-k$ являются генераторами стабилизатора, но есть $2k$ дополнительных независимых операторов ошибок, не принадлежащих стабилизатору, но коммутирующих с ним. Эти $2k$ операторов сохраняют кодовое подпространство, но действуют нетривиально на кодовые слова и, следовательно, их можно интерпретировать как операции, действующие на закодированные логические кубиты.

Собственно, эти $2k$ операторов можно выбрать в качестве однокубитовых операций $\hat{Z}_i$ и $\hat{X}_i$, где $i = 1, 2, 3, \ldots k$ нумерует закодированные кубиты. Сначала отметим, что $n-k$ генераторов стабилизатора можно расширить до максимального набора n коммутирующих операторов; k дополнительных операторов можно идентифицировать как операторы $\hat{Z}_i$. Мы можем выбрать состояния вычислительного базиса в кодовом подпространстве таким образом, чтобы они одновременно были собственными состояниями всех операторов $\hat{Z}_i$, с собственным значением $+1$, соответствующим значению 0, и собственным значением -1 соответствующим значению 1. Тогда $\hat{Z}_i$ обращает фазу i-го кубита. Остальные k генераторов $\hat{X}_i$, которые коммутируют со стабилизатором, но не коммутируют со всеми $\hat{Z}_i$, можно выбрать так, чтобы они удовлетворяли соотношениям

$$[\hat{Z}_i, \hat{Z}_j] = 0 = [\hat{X}_i, \hat{X}_j], \quad [\hat{Z}_i, \hat{X}_j] = 0, \ (i \neq j), \qquad \hat{Z}_i\hat{X}_i + \hat{X}_i\hat{Z}_i = 0. \quad (28)$$

Так как $\hat{X}_i$ антикоммутирует с $\hat{Z}_i$, он обращает собственное значение $\hat{Z}_i$ и, следовательно, значение i-го кубита. Все эти операции выполняются с применением максимум одного однокубитового вентиля к каждому кубиту в блоке; следовательно, эти операции без сомнения отказоустойчивые.

Мы также видели в разделе 3.6, что возможно выполнение отказоустойчивого *измерения любого* оператора $\bar{Z}\bar{X}$, а значит, в частности, возможно отказоустойчивое измерение любого оператора ошибки $\hat{X}_i$, $\hat{Y}_i$, $\hat{Z}_i$. Готтесман [17] показал, что если возможно выполнение вентиля Тоффоли

(который является универсальным для *классических* вычислений, сохраняющих набор состояний вычислительного базиса), то для универсальных квантовых вычислений достаточно однокубитовых вентилей X и Z вместе с возможностью измерения X, Y и Z для любого кубита. Следовательно, если мы покажем, что, действуя на любые три кубита, можно построить отказоустойчивый вентиль Тоффоли, то этим завершим доказательство возможности универсальных отказоустойчивых квантовых вычислений для любого стабилизирующего кода.

Конструкция отказоустойчивого вентиля Тоффоли достаточно сложна, поэтому доказательство лучше построить следующим образом: Готтесман показал, что в любом стабилизирующем коде отказоустойчивый XOR-вентиль может применяться к любой паре кубитов (независимо от того, находятся или нет эти два кубита в одном и том же кодовом блоке). Используя XOR-вентиль, а также однокубитовые вентили и измерения, которые, как мы только что видели, можно выполнять отказоустойчиво, можно построить все вентили, необходимые в конструкции Шора вентиля Тоффоли. Таким образом, с помощью любого стабилизирующего кода можно реализовать отказоустойчивую схему вентиля Тоффоли, что достаточно для выполнения универсальных отказоустойчивых квантовых вычислений.

Несмотря на то, что отказоустойчивые квантовые вычисления, в принципе, можно реализовать с помощью любого стабилизирующего кода, некоторые из них лучше подходят для этой цели. Например, существует 5-кубитовый код, способный исправить одну ошибку [36,37]; Готтесман продемонстрировал универсальный набор отказоустойчивых вентилей для этого кода. Но их реализация достаточно сложна. 7-кубитовый код Стина требует блок большего размера, но он гораздо удобнее для вычислений.

5. Порог безошибочности квантовых вычислений

Существуют квантовые коды, способные исправить t ошибок, где t может быть сколь угодно большим. Если мы используем такой код и придерживаемся принципов отказоустойчивости, то непоправимая ошибка возникнет только при появлении минимум $t+1$ независимых ошибок в одном блоке до завершения процесса исправления. Поэтому если вероятность возникновения ошибки на квантовый вентиль или вероятность возникновения ошибки хранения на единицу времени имеет порядок ϵ, то вероятность появления ошибки в расчете на один вентиль, действующий на закодированные данные, будет иметь порядок ϵ^{t+1}, что гораздо меньше, чем ϵ, если эта величина достаточно мала. Действительно, может показаться, что при выборе кода со сколь угодно большим t мы можем сделать вероятность воз-

никновения ошибки в расчете на один вентиль сколь угодно малой, но, оказывается, это не так, по крайней мере, не для большинства кодов. Проблема в том, что по мере увеличения t сложность кода резко возрастает, и соответственно возрастает сложность процедуры исправления. В конечном счете, мы достигаем момента, когда выполнение исправления занимает так много времени, что становится возможным накопление $t+1$ ошибок в блоке до полного завершения этапа исправления, и, таким образом, способность кода исправлять ошибки становится сомнительной.

Предположим, что количество вычислительных этапов, необходимых для выполнения измерения синдрома, растет вместе с t, как степень t^b. Тогда вероятность того, что до завершения измерения накопится $t+1$ ошибок, будет вести себя как

$$\text{Block Error Probability} \sim (t^b \epsilon)^{t+1}, \tag{29}$$

где ϵ — вероятность появления ошибки на одном этапе. Тогда мы можем выбрать t, чтобы минимизировать вероятность ошибки ($t \sim e^{-1}\epsilon^{-1/b}$, при большом t), и получить

$$\text{Minimum Block Error Probability} \sim \exp\left(e^{-1} b \epsilon^{-1/b}\right). \tag{30}$$

Таким образом, если мы рассчитываем безукоризненно выполнить все T циклов исправления ошибок, то наши вентили должны иметь точность

$$\epsilon \sim (\log T)^{-b}. \tag{31}$$

Аналогично, для выполнения квантового вычисления с участием T квантовых вентилей необходимы элементарные вентили заданной точности.

В первоначально описанной Шором процедуре [15] степень, характеризующая сложность измерения синдрома, равна $b = 4$; с помощью более оптимизированной процедуры можно достичь несколько меньших значений b. Размер блока используемого кода растет вместе с t как t^2 (для рассмотренных Шором кодов), поэтому при выборе кода для оптимизации вероятности появления ошибки, размер блока имеет порядок $(\log T)^2$. Конечно, описываемый уравнением (31) скейлинг гораздо предпочтительнее, чем точность $\epsilon \sim T^{-1}$, которая потребовалась бы, если бы кодирование не использовалось вообще. Но при любой заданной точности существует предел продолжительности вычисления, при которой еще можно не опасаться появления ошибок.

Это ограничение можно преодолеть, используя код специального типа — *каскадный* код [18–24]. Чтобы понять идею каскадного кода, представим, что мы используем исправляющий ошибки квантовый код Стина,

кодирующий единственный кубит в 7-кубитовом блоке. Но если мы внимательнее рассмотрим один из семи кубитов в блоке, то обнаружим, что на самом деле это не один кубит, а другой блок из семи закодированных с помощью того же кода Стина кубитов. А когда мы с еще бóльшим разрешением исследуем один из семи кубитов в уже этом блоке, мы обнаружим, что он также является блоком из семи кубитов, и так далее (см. рис. 14). Если в этой иерархии каскадного соединения всего имеется L уровней, то отдельный кубит фактически кодируется в блоке размера 7^L. Каскадное соединение оказывается полезным, поскольку, действуя по принципу «разделяй и властвуй», оно позволяет более эффективно избавляться от ошибок. При таком методе сложность этой процедуры уже не так стремительно растет с повышением способности квантового кода исправлять ошибки.

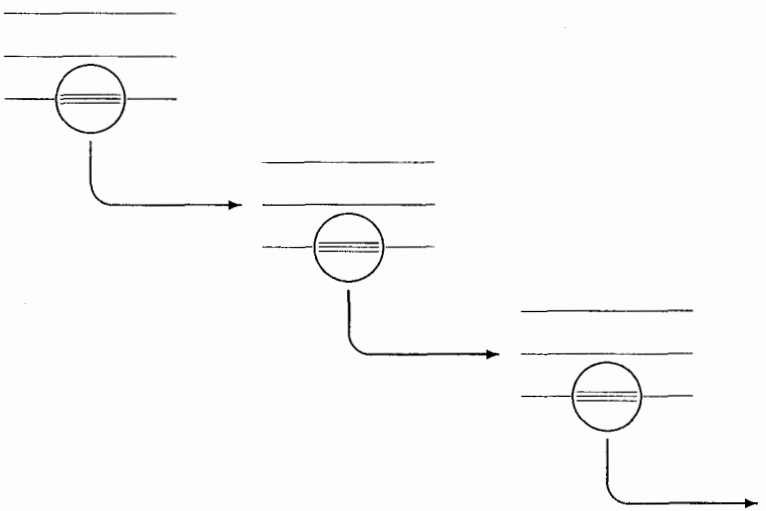


Рис. 14. Каскадное кодирование. Каждый кубит в блоке, при ближайшем рассмотрении, сам по себе является закодированным субблоком

Мы убедились, что 7-кубитовый код Стина может исправить одну ошибку. Если вероятность возникновения ошибки на кубит равна ϵ, ошибки некоррелированы, а исправление отказоустойчиво, тогда вероятность отказа при исправлении имеет порядок ϵ^2. Если два кода соединяются в каскад, образуя блок размера 7^2, то ошибка в нем возникает только при повреждении двух из его субблоков размера семь, что происходит с вероятностью порядка $(\epsilon^2)^2$. При добавлении еще одного уровня каскадирования

ошибка в блоке размера 7^3 возникает только в случае повреждения двух из его субблоков размера 7^2, а вероятность такого события имеет порядок $\left((\epsilon^2)^2\right)^2$, и так далее. При наличии L уровней каскадного соединения кодов вероятность появления ошибки имеет порядок ϵ^{2^L}, тогда как размер блока равен 7^L. Если частота появления ошибок в основных вентилях достаточно мала, то отнесенную к одному вентилю вероятность появления ошибки можно уменьшить с помощью каскадного соединения кодов. Если это так, то добавление следующего уровня каскадирования приведет к дальнейшему понижению вероятности ошибки, и так далее. В этом состоит природа порога безошибочности квантовых вычислений: если кодирование значительно снижает вероятность появления ошибки, то, добавляя достаточное количество уровней каскадирования, частоту появления ошибок можно сделать сколь угодно малой. Но если частоты появления ошибок изначально слишком высоки, тогда вместо улучшения кодирование только усугубит существующее положение вещей.

Чтобы проанализировать эту ситуацию, необходимо принять некоторую конкретную модель ошибок. Я выберу наиболее простую из возможных квазиреалистическую модель: некоррелированые стохастические ошибки.[1] На каждом такте вычислений каждый кубит в устройстве запутывается с окружением. Пусть запутывание описывается уравнением (4) с тем лишь отличием, что теперь четыре состояния окружения предполагаются взаимно ортогональными, а «ошибочные состояния» — имеющими одинаковые нормы. Таким образом, три типа ошибок (инвертирование бита, обращение фазы, обе одновременно) предполагаются равновероятными. Полная вероятность ошибки на каждом такте вычислений обозначается как ϵ_{store}. Помимо этих ошибок запоминающего устройства, повреждающих хранящиеся кубиты, существуют ошибки, вносимые самими квантовыми вентилями. Для каждого типа вентиля вероятность появления ошибки при каждом его выполнении обозначается ϵ_{gate} (при независимых значениях, приписываемых вентилям каждого типа). Если вентиль действует более чем на один кубит (XOR или Тоффоли), то могут возникнуть коррелированые ошибки. Сделаем пессимистическое допущение, что ошибка многокубитового вентиля всегда повреждает все кубиты, на которые он действует; например, неправильный XOR-вентиль вносит ошибки как в кубит источника, так и в кубит цели. Это предположение (среди прочих) сделано лишь для упрощения анализа. При более реалистичных допущениях мы обнаружили бы, что вполне можно пережить и несколько более высокие частоты появления ошибок.

[1] Более детальную характеристику модели ошибок я приведу в разделе 6.

Эффективность каскадного кодирования можно проанализировать, построив систему *потоковых уравнений*, описывающих эволюцию модели ошибки при переходе от одного уровня каскадирования к другому. Пусть, например, мы хотим выполнить XOR-вентиль с последующим этапом исправления ошибок в кубитах, закодированных с помощью каскадного кода Стина с L уровнями каскадирования (размер блока 7^L). Эти процедуры можно описать с помощью таких же операций, но действующих на субблоки размера 7^{L-1}. Таким образом, вероятность появления ошибки $\epsilon^{(L)}$ для вентиля, действующего на блок размера L, может быть выражена через вероятность появления ошибки $\epsilon^{(L-1)}$ для вентиля, действующего на блок размера $L-1$. Это соотношение представляет собой одно из потоковых уравнений. В принципе, решая систему потоковых уравнений, можно получить выражение для вероятности появления ошибки «на уровне L» через параметры модели ошибок и исследовать ее поведение с ростом L. Если вероятность появления ошибки в блоке стремится к нулю с ростом его размера L, то это означает, что вероятности элементарных ошибок лежат «ниже порога». Так как вероятности элементарных ошибок могут зависеть от множества параметров, то на самом деле порог представляет собой некоторую гиперповерхность в многомерном параметрическом пространстве рассматриваемой модели.

Метод Стина измерения синдрома очень хорошо подходит для каскадного кодирования. Все вентили корректирующей схемы (рис. 9) можно выполнять *трансверсально*; выполнение вентилей на элементарных кубитах кодового блока означает применение точно таких же вентилей к информации, закодированной в каждом блоке размера семь, в каждом суперблоке размера 7×7, и так далее. Аналогично, когда при завершении вычисления синдрома мы измеряем элементарные кубиты в служебном блоке, тогда (после применения к кубитам классической коррекции ошибок Хэмминга) мы также измеряем кубиты, закодированные в каждом блоке размера семь, и (после применения коррекции ошибок Хэмминга к блокам) в каждом суперблоке размера 7×7, и так далее. Таким образом, необходимая для извлечения синдрома обработка квантовых данных может быть выполнена *одновременно на всех уровнях каскадного кода*.[1] После чего результаты некоторой классической обработки указывают, какие однокубитовые вентили необходимо применить ко всем элементарным кубитам, чтобы завершить этап исправления на всех уровнях одновременно.

[1] *Разрушающее* измерение закодированного служебного блока можно выполнить одновременно на всех уровнях. Процедура *неразрушающего* измерения блока (проецирование блока на $|0\rangle_{\text{code}}$ или $|1\rangle_{\text{code}}$) гораздо более трудоемка; в каждый момент времени она должна выполняться на одном уровне.

Таким образом, нетрудно понять, как (по крайней мере, в принципе) можно оценить порог безошибочности [38]. На каждом уровне каскадного кода блок из семи кубитов отказывает, если ошибки возникают как минимум в двух его субблоках. Если p_L — вероятность появления ошибки в блоке на уровне L, то вероятность ошибки на уровне $L+1$ равна

$$p_{L+1} \sim \binom{7}{2} p_L^2 + \ldots = 21 p_L^2 + \ldots \tag{32}$$

(с точностью до слагаемых более высокого порядка по p_L), что будет меньше p_L при $p_L < 1/21$. Следовательно, если вероятность появления ошибки в каждом элементарном кубите $p_0 < 1/21$, то на первом уровне кодирования она будет меньше этой величины, еще меньше — на втором уровне и так далее — значение порога безошибочности p_0 равно 1/21.

Предположим, мы выполняем исправление ошибок каждый раз при осуществлении XOR-вентиля или однокубитового вентиля. Грубо говоря, p_0 — это вероятность ошибки на кубит информации в момент начала цикла исправления ошибок. Для оценки порога безошибочности мы следуем схеме на рис. 9 и увеличиваем вследствие появления ошибок значение p_0 (включая возможные ошибки хранения), которые возникли при только что выполненных квантовых вентилях и еще не были устранены предыдущим циклом исправления ошибок. Мы получаем выражение для p_0 через вероятности ошибки вентиля и ошибки хранения; это выражение приравнивается к $1/21$ для определения порога.

Поступая таким образом и предполагая, что ошибки запоминающего устройства пренебрежимо малы, а все однокубитовые и XOR-вентили имеют одинаковую вероятность появления ошибки ϵ_{gate}, мы [38] грубо оцениваем порог частоты ошибок вентиля как

$$\epsilon_{\text{gate},0} \sim 6 \cdot 10^{-4}. \tag{33}$$

Аналогично, если ошибки вентиля пренебрежимо малы, предполагаемый порог частоты ошибок запоминающего устройства равен

$$\epsilon_{\text{store},0} \sim 6 \cdot 10^{-4}. \tag{34}$$

Пороги для ошибок вентиля и запоминающего устройства по существу одинаковы, так как метод Стина хорошо приспособлен для работы с ошибками хранения. Кубиты редко лежат без дела; практически на каждом этапе на каждый из них действует вентиль. Следовательно, требование к точности запоминающего устройства значительно менее жесткое по сравнению с предыдущими оценками порога, основанными на методе исправления Шора [23, 35, 39].

Однако, как показывает более тщательный анализ, по ряду причин фактическое значение порога, которое можно вывести из схемы на рисунке 9, несколько ниже оценок (33) и (34). Наиболее серьезное требование состоит в том, что для осуществления исправления необходим запас закодированных на уровне L хорошо проверенных состояний $|0\rangle_{\text{code}}$. Отдельное (и весьма сложное) вычисление требуется для определения порога надежного кодирования. Кроме того, необходимо проанализировать реализацию Шора вентиля Тоффоли, чтобы гарантировать возможность его надежного применения к соединенным в каскад блокам.[1] Наконец, для получения точного результата необходимо ограничить вклады высшего порядка в вероятность отказа, которые были опущены в уравнении (32). Полный анализ, учитывающий все эти факторы, еще не выполнен, но вполне разумно предположить, что конечные величины порогов запоминающего устройства и вентиля будут превышать 10^{-4}. Конечно, при более совершенной кодирующей схеме и/или протоколе исправления ошибок, возможно, удастся достичь гораздо более высокого значения порога безошибочности.

Также следует задать вопрос, какой размер блока необходим для обеспечения некоторой заданной точности. Грубо говоря, если пороговая частота ошибок вентиля равна ϵ_0, а фактическая частота ошибок элементарного вентиля равна $\epsilon < \epsilon_0$, то L-кратное каскадирование кода понижает частоту появления ошибок до

$$\epsilon^{(L)} \sim \epsilon_0 \left(\frac{\epsilon}{\epsilon_0}\right)^{2^L}. \tag{35}$$

Таким образом, чтобы быть достаточно уверенными в том, что мы можем безошибочно завершить вычисление при T вентилях, выбранный нами размер блока 7^L должен иметь порядок

$$\begin{matrix}\text{block}\\ \text{size}\end{matrix} \sim \left[\frac{\log \epsilon_0 T}{\log \epsilon_0/\epsilon}\right]^{\log_2 7}. \tag{36}$$

Если каскадный код имеет размер блока n и может исправить $t+1$ ошибок, степень $\log_2 7 \sim 2.8$ в уравнении (36) заменяется на $\log n/\log(t+1)$; эта степень приближается к 2 для рассмотренных Шором семейств кодов, но для «хороших» кодов, в принципе, может приблизиться к 1.

Когда частоты появления ошибок находятся ниже порога безошибочности, также возможно сколь угодно продолжительное хранение *неизвест-*

[1] Элементарные вентили Тоффоли не должны быть такими же точными, как одно- и двухкорпусные вентили — вполне приемлема частота появления ошибок вентиля Тоффоли порядка 10^{-3}, если достаточно низки частоты других ошибок. Этот вывод приятен, поскольку вентили Тоффоли сложнее в применении и на практике, вероятно, имеют меньшую точность.

ного квантового состояния. Однако, как мы уже отмечали в разделе 3.5, если вероятность возникновения ошибки запоминающего устройства на один такт вычисления равна ϵ, то первоначальное кодирование состояния можно выполнить с точностью воспроизведения, не превышающей $F = 1 - O(\epsilon)$. При каскадном кодировании мы можем сколь угодно долго хранить неизвестную квантовую информацию с достаточно высокой точностью воспроизведения, но достичь сколь угодно высокой точности воспроизведения не можем.

Каскадирование — важная теоретическая конструкция, так как она позволяет утверждать, что возможны сколь угодно продолжительные вычисления. Но пока частоты ошибок достаточно далеки от пороговых значений, каскадное кодирование, возможно, будет не лучшим способом выполнения определенного вычисления заданной длины. Действительно, выбранный из первоначально описанного Шором семейства код может оказаться более эффективным, чем каскадный 7-битовый код. Более того, каскадный 7-битовый код и коды Шора кодируют лишь один кубит квантовой информации в достаточно большом кодовом блоке. Но в разделе 4 мы видели, что отказоустойчивые квантовые вычисления можно выполнять, используя любые стабилизирующие коды, в том числе и те, что путем кодирования множества кубитов в одном блоке делают более эффективным использование объема памяти. Если надежность наших аппаратных средств близка к порогу безошибочности, то эти коды будут работать неэффективно. Но с усовершенствованием «железа» можно использовать более эффективные коды и таким образом повышать надежность нашего квантового компьютера при меньших затратах объема памяти.

6. Модели ошибок

Отказоустойчивая схема должна быть приспособлена для защиты от тех типов ошибок, которые могут с большей вероятностью нанести ущерб конкретному устройству. И любое утверждение о величине допустимых частот возникновения ошибок (как и оценка порога безошибочности, набросок которой мы только что сделали) бессмысленно, пока не будет тщательным образом определена модель ошибок. Подытожим некоторые важные предположения о характере ошибок, лежащие в основе нашей оценки порога безошибочности.

- **Случайные ошибки.** Мы предположили, что ошибки не имеют систематической составляющей.[1] Ошибки, имеющие случайные фазы,

[1] Нилл и др. [19] показали существование порога безошибочности для гораздо более общих моделей ошибок.

накапливаются по сценарию случайного блуждания, так что с количеством применяемых вентилей приблизительно линейно растет *вероятность* ошибки. Но если ошибки имеют систематические фазы, тогда линейно с числом применяемых вентилей может расти *амплитуда* вероятности ошибки. Следовательно, чтобы наш квантовый компьютер работал хорошо, частота систематических ошибок должна удовлетворять более жестким требованиям, нежели частота случайных ошибок. Иначе говоря, если мы допускаем, что систематические фазы тайно договариваются всегда складываться конструктивно, и если для случайных ошибок порог безошибочности равен ϵ_0, то для (максимально законспирированных) систематических ошибок он будет иметь порядок ϵ_0^2. Несмотря на то, что систематические ошибки могут стать проблемой для квантовых инженеров будущего, они не должны представлять собой непреодолимое препятствие. Моя позиция состоит в следующем: (1) даже если наши аппаратные средства предрасположенные к появлению ошибок с систематическими фазами, эти ошибки будут стремиться к взаимному уничтожению в ходе достаточно продолжительного вычисления [40–42], и (2) так как систематические ошибки можно, в принципе, понять и устранить, с фундаментальной точки зрения более важно иметь представление об ограничениях, накладываемых на работу машины случайными ошибками.

- **Некоррелированные ошибки.** Мы предположили, что ошибки не коррелируют как в пространстве, так и во времени. Таким образом, когда мы говорим, что вероятность возникновения ошибки на кубит равна (например) $\epsilon \sim 10^{-5}$, то фактически имеем в виду, что для двух заданных кубитов вероятность одновременного повреждения ошибками обоих равна $\epsilon^2 \sim 10^{-10}$. Это очень сильное предположение. Действительно важное условие состоит в том, чтобы коррелированные ошибки, повреждающие множество кубитов в одном и том же кодовом блоке, были в высшей степени маловероятны, так как кодирующие схемы будут отказывать при появлении нескольких ошибок в одном блоке. Квантовые инженеры будущего столкнутся с проблемой конструирования таких устройств, в которых кубиты одного блока были бы тщательно изолированы друг от друга.

- **Максимальный параллелизм.** Мы предположили, что в течение одного такта может параллельно выполняться несколько квантовых вентилей. Это допущение позволяет выполнять исправление ошибок во всех кодовых блоках одновременно, и поэтому важно для контроля ошибок хранения кубитов. (В противном случае, добавление к коду

следующего уровня каскадирования вело бы к росту вероятности отказа, поскольку каждому не занятому в процессе отдельному кубиту пришлось бы дольше ожидать своей очереди исправления ошибок.) Если мы пренебрегаем ошибками запоминающего устройства, то для анализа порога безошибочности параллельность выполнения операций не обязательна, но для ускорения вычислений она, безусловно, желательна.

- **Независимая от числа кубитов частота появления ошибок.** Мы предположили, что частоты ошибок не зависят от количества хранящихся в нашем устройстве кубитов. Неявно это допущение касается природы аппаратных средств. Например, это предположение было бы необоснованным, если бы все кубиты хранились в единственной ионной ловушке и делили бы один фононный канал передачи информации [43].
- **Вентили могут действовать на любую пару кубитов.** Мы предположили, что наша машина снабжена набором базовых вентилей, которые можно применить к любой паре хранящихся кубитов (или тройке кубитов, в случае вентиля Тоффоли), независимо от их близости друг к другу. На практике возможны издержки как по времени выполнения, так и по частоте появления ошибок, связанные с перемещением кубитов для того, чтобы вентиль мог эффективно действовать на конкретную пару. Выбор архитектуры, минимизирующей эти издержки, мы оставим конструкторам машин. При наличии вентилей, действующих только на соседние кубиты, порог по-прежнему будет существовать [21], но он станет значительно ниже.
- **Новые служебные кубиты.** Мы предположили, что наш компьютер имеет доступ к достаточному запасу новых служебных кубитов. Служебные кубиты используются как для выполнения вентилей (Тоффоли), так и для осуществления исправления ошибок. Вместе с накоплением эффектов случайных ошибок генерируется энтропия, а процесс исправления ошибок выбрасывает ее из вычислительного устройства в служебный регистр. В принципе, пока поставляются свежие служебные кубиты, вычисление может продолжаться сколь угодно долго, но на практике мы захотим очищать служебные кубиты и использовать их повторно. Стирание служебного кубита неизбежно вызовет рассеяние мощности и выделение тепла; поэтому потребуется охлаждение устройства.
- **Отсутствие ошибок утечки.** Мы пренебрегли вероятностью утечки. В нашей модели квантового компьютера каждый из кубитов живет

в двумерном гильбертовом пространстве, и мы предполагаем, что при появлении ошибки этот кубит либо запутывается с окружающей средой, либо поворачивается в двумерном пространстве в непредсказуемом направлении. Но существует и другой возможный тип ошибки, при которой кубит просачивается из двумерного в более широкое пространство [44]. Чтобы контролировать ошибки утечки, мы можем повторно запрашивать каждый кубит (например, используя показанную на рисунке 15 схему определения места утечки), не пытаясь точно диагностировать, что произошло с просочившимся кубитом [23]. При возникновении утечки кубит разрушается, и его необходимо отбросить;[1] мы заменяем его свежим кубитом в стандартном состоянии, скажем, в состоянии $|0\rangle$. Затем мы можем выполнить стандартное измерение синдрома, которое спроецирует этот кубит на такое состояние, что ошибку можно будет исправить с помощью простого унитарного преобразования.[2] При использовании каскадного кодирования детектирование утечки должно выполняться только на самых нижних уровнях кодирования. Схема определения достаточно проста, поэтому наличие ошибок утечки лишь незначительно повлияет на порог безошибочности.

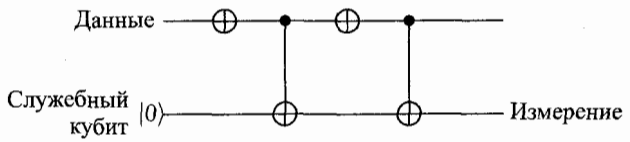


Рис. 15. Квантовая схема детектирования утечки. Допустим, что при утечке данных XOR-вентиль действует тривиально, тогда при возникновении утечки результат измерения равен 0, в противном случае — 1

Предположения нашей модели ошибок в достаточной степени реалистичны, чтобы обеспечить разумную оценку того, насколько хорошо может работать квантовый компьютер при наличии шума. Предположим, например, мы хотим, чтобы наш квантовый компьютер решил сложную задачу факторизации с использованием алгоритма Шора; каким техническим требованиям должна отвечать машина? Располагая самым известным класси-

[1] Конечно, мы можем впоследствии использовать его повторно.

[2] Действительно, так как еще до измерения синдрома мы знаем, что поврежденный кубит находится в определенной позиции в кодовом блоке, для диагностики и коррекции ошибки в известном положении мы можем применить ускоренную версию исправления ошибок [45].

ческим алгоритмом факторизации и самой быстродействующей существующей машиной, факторизацию 130-разрядного (432-битового) числа можно выполнить за несколько месяцев [46]. Чтобы решить эту задачу с помощью алгоритма Шора, мы должны быть в состоянии хранить около $5 \cdot 432 = 2160$ кубитов и выполнить около $38 \cdot (432)^3 \sim 3 \cdot 10^9$ вентилей Тоффоли [47]. Чтобы иметь достаточно шансов выполнить такое вычисление с приемлемой точностью, необходимо, чтобы вероятность ошибки на вентиль Тоффоли была меньше 10^{-9}, а вероятность ошибки запоминающего устройства на время выполнения вентиля — меньше 10^{-12}.

Согласно потоковым уравнениям каскадирования для 7-кубитового кода,[1] для закодированных данных эти частоты возникновения ошибок могут быть достигнуты, если частоты ошибок на уровне отдельных кубитов равны $\epsilon_{\text{store}} \sim \epsilon_{\text{gate}} \sim 10^{-6}$ и если используется три уровня каскадирования, так что размер блока, кодирующего каждый кубит, равен $7^3 = 343$. С учетом дополнительных служебных кубитов, необходимых для выполнения вентилей и (параллельного) исправления ошибок, требуемое машине общее число кубитов будет иметь порядок 10^6.

При достаточно высокой частоте появления ошибок запоминающего устройства каскадирование может стать наиболее эффективной кодирующей процедурой. Но если доминируют ошибки вентилей (и если частота их появления не слишком близка к пороговой), тогда лучше будут работать другие коды. Например, Стин [48] обнаружил, что с использованием кода с размером блока 55, способного исправить пять ошибок, эта задача факторизации может быть решена квантовым компьютером при наличии $4 \cdot 10^5$ кубитов и частоте появления ошибок вентиля порядка 10^{-5}. При более низких частотах появления ошибок можно применять коды, более эффективно использующие объем памяти путем кодирования множества кубитов в одном блоке [17].

Несомненно, квантовый компьютер, содержащий около миллиона кубитов и имеющий отнесенную к одному вентилю частоту появления ошибки около одной на миллион, был бы очень мощным и ценным устройством (при достаточной скорости обработки). Конечно, с точки зрения текущего состояния технологии [49–52], эти числа выглядят удручающими. Но даже машина, удовлетворяющая гораздо менее жестким техническим требованиям, может быть очень полезной [53]. Во-первых, помимо факторизации квантовые компьютеры способны на многое другое, и некоторые из этих задач (в частности, квантовое моделирование [54]) можно выполнить с по-

[1] Этот анализ [23] был выполнен скорее для измерения синдрома методом Шора, нежели методом Стина, на который мы ссылались в разделе 5 при обсуждении каскадного кодирования.

мощью менее надежного или меньшего по размеру устройства. Более того, наша оценка порога безошибочности, возможно, по ряду причин слишком консервативна. В частности, она была получена в предположении, что фазовые и амплитудные ошибки в кубитах равновероятны. Располагая более реалистичной моделью, лучше представляющей вероятности ошибок в данном устройстве, к ней можно было бы лучше приспособить схему исправления ошибок и, следовательно, пережить более высокую частоту появления ошибок. Но даже при сформулированных допущениях приведенный анализ отказоустойчивой схемы не вполне точен; при более тонком анализе можно ожидать обнаружение несколько более высокого порога безошибочности, возможно, значительно более высокого. Также значительных усовершенствований можно достичь путем модификации схемы отказоустойчивости, или с помощью новых, более эффективных, способов применения универсального набора отказоустойчивых вентилей или средств выполнения измерений синдрома ошибки. С учетом различных усовершенствований уже не кажется удивительным то, что квантовый компьютер сможет эффективно работать при вероятности появления ошибок в расчете на один вентиль, например, порядка 10^{-3}.[1]

Конечно, частота появления ошибок, скажем, 10^{-5} весьма претенциозна, но, наверное, эта величина не лежит за рамками возможно достижимого в будущем. В любом случае, сейчас мы имеем четкое представление о степени надежности работы полезного квантового компьютера, что само по себе представляет невероятный прогресс по сравнению с тем, что мы имели всего лишь два года назад.

7. Топологические квантовые вычисления

7.1. Эффект Аронова–Бома и правила суперотбора

Теперь, когда мы не сомневаемся в возможности исправления квантовых ошибок, важно взглянуть на эту проблему шире, а именно, попытаться выйти за рамки анализа абстрактных схем и исследовать потенциальные физические условия, в которых можно надежно хранить и обрабатывать квантовую информацию. В частности, мы могли бы рассчитывать на создание *внутренне* отказоустойчивых квантовых вентилей, не требующих активного вмешательства оператора вычислительной машины для защиты ее от шума. Важный шаг в направлении к этой цели был недавно сделан Алексеем Китаевым [25], настоящий раздел основан на его идеях.

[1] Действительно, Залкой были предложены более оптимистичные по сравнению с моими оценки порога безошибочности [24].

Топологические идеи естественным образом возникают при обсуждении коррекции квантовых ошибок и отказоустойчивых вычислений. Топология интересуется «глобальными» свойствами объекта, которые остаются неизменными при его локальной деформации. Основная идея коррекции квантовых ошибок включает хранение и обработку квантовой информации в «глобальной» форме, устойчивой к локальным возмущениям. Конструкция отказоустойчивого вентиля должна позволять ему действовать на эту глобальную информацию так, чтобы направленное им на закодированные данные действие оставалось неизменным даже при некоторой деформации данного вентиля, то есть даже при его неидеальном выполнении.

Пытаясь найти физическую реализацию отказоустойчивых квантовых вычислений, зададимся вопросом: существуют ли системы, в которых физические взаимодействия имеют топологическую природу? Несомненно, топология лежит в основе *эффекта Аронова–Бома*. Если электрон обходит вокруг идеально заэкранированного магнитного соленоида, его волновая функция приобретает фазу $e^{ie\Phi}$, где e — заряд электрона, а Φ — запертый внутри соленоида магнитный поток. Эта фаза Аронова–Бома является топологическим свойством пройденного электроном пути — она зависит лишь от количества оборотов вокруг соленоида и остается неизменной при непрерывной деформации траектории обхода (см. рис. 16). Это заставляет нас подумать о такой реализации квантовых вычислений, в которой закодированная информация могла бы измеряться и обрабатываться с помощью взаимодействий Аронова–Бома — устойчивых к локальным возмущениям взаимодействий топологической природы.

Полезно выразить эти доводы еще раз на языке правил суперотбора. Правило суперотбора, как я использую здесь этот термин, возникает (в теории поля или спиновой системы, определенной в бесконечном пространственном объеме), если гильбертово пространство распадается на взаимно ортогональные секторы, каждый из которых сохраняется под действием любой локальной операции. Пожалуй, самым известным примером является правило суперотбора заряда в квантовой электродинамике. Электрическое поле, создаваемое зарядом, имеет бесконечный радиус действия. Следовательно, никакая локальная операция не в состоянии создать или уничтожить заряд. Действительно, для этого необходимо создать или уничтожить простирающиеся в бесконечность силовые линии электрического поля, но ни одна локальная процедура не может справиться с этой задачей.

Взаимодействие Аронова–Бома также имеет бесконечный радиус действия; по мере обхода вокруг соленоида электрон приобретает фазу Аронова–Бома независимо от их взаимной удаленности. Поэтому можно сказать, что ни одна локальная операция не может уничтожить заряд, при-

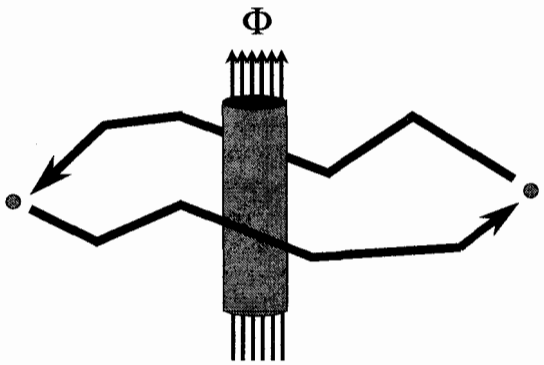


Рис. 16. Топологические взаимодействия. Фаза Аароновa–Бома, приобретаемая электроном при обходе трубки потока, остается неизменной при непрерывной деформации его пути

нимающий участие в явлении Аароновa–Бома. Если мы рассмотрим два несущих такие заряды объекта, находящихся на значительном расстоянии друг от друга и хорошо изолированных от других заряженных объектов, то любой процесс, изменяющий заряд на одном из этих двух объектов, должен был бы действовать когерентно во всей содержащей их области. Таким образом, в присутствии локальных возмущений заряды достаточно устойчивы; мы можем ударить по частице молотком или повредить ее любым другим способом, но переносимый ею заряд мы этим не изменим.

Следуя Китаеву [25], мы можем представить себе *топологический квантовый компьютер* — устройство, в котором квантовая информация кодируется в квантовых числах, переносимых квазичастицами, лежащими на двумерной плоскости и влияющими друг на друга посредством дальнодействующего взаимодействия Аароновa–Бома. При нулевой температуре случайная перестановка квантовых чисел между квазичастицами (ошибка) возникает только из-за явления квантового туннелирования, влекущего за собой виртуальную перестановку заряженных объектов. Амплитуда такого процесса имеет порядок e^{-mL}, где m — масса самого легкого заряженного объекта (в естественных единицах), а L — расстояние между двумя квазичастицами. Если квазичастицы удерживаются на достаточно большом расстоянии друг от друга, вероятность появления ошибки, поражающей закодированную информацию, будет чрезвычайно низкой. При конечной температуре T появляется дополнительный источник ошибок, обусловленный

неизбежной при $T > 0$ генерацией плазмы заряженных частиц с плотностью, пропорциональной фактору Больцмана $e^{-\Delta/T}$, где Δ — массовая щель (не обязательно равная «массе кривизны» m). Иногда одна из частиц плазмы может проскользнуть незамеченной между двумя нашими частицами — носителями данных, что приведет к перестановке зарядов и, следовательно, к ошибке. Итак, для достижения приемлемо низкой частоты появления ошибок необходимо поддерживать уровень температуры значительно ниже щели Δ (в противном случае нам пришлось бы тщательно контролировать термически возбужденную плазму).

7.2. Дробный квантовый эффект Холла (и не только)

Для того чтобы наше устройство было способно выполнять интересные вычисления, применяемые им явления Аронова–Бома должны быть *неабелевыми*. Только в этом случае можно построить сложные унитарные преобразования, выполняя множество следующих друг за другом перестановок частиц. Такие неабелевы эффекты Аронова–Бома могут возникнуть в системах с неабелевыми калибровочными полями. Природа оказалась весьма благосклонной — она снабдила нас некоторыми фундаментальными неабелевыми калибровочными полями, но, к сожалению, не слишком большим количеством, и, похоже, ни одно из них не подходит для практических квантовых вычислений. В таком случае нам остается надеяться на то, что подходящие для реализации идеи Китаева неабелевы эффекты Аронова–Бома могут возникать как сложные кооперативные явления в (двумерных электронных или спиновых) системах, в которых существуют лишь короткодействующие фундаментальные взаимодействия.

Тот факт, что дальнодействующие явления Аронова–Бома могут возникать в таких системах (как это показали наблюдения дробного квантового эффекта Холла), является одним из наиболее замечательных открытий последних десятилетий. Электронная система в режиме квантового эффекта Холла настолько фрустрирована, что ее основное состояние представляет собой в высшей степени запутанное состояние с простирающимися на большие расстояния сильными квантовыми корреляциями. Следовательно, когда одна квазичастица обходит вокруг другой, даже если они расположены на большом расстоянии друг от друга, многоэлектронная волновая функция приобретает нетривиальную фазу Берри (такую как $e^{2\pi i/3}$). Эта фаза Берри во всех ее наблюдаемых проявлениях неотличима от фазы Аронова–Бома, происходящей от фундаментального калибровочного поля, а ее экспериментальные последствия впечатляющи [55].

Наблюдаемые в режиме квантового эффекта Холла фазы Берри — абелевы (хотя имеются некоторые серьезные признаки того, что при соответ-

ствующих условиях могут возникать неабелевы фазы Берри [56,57]) и, следовательно, не особенно интересны с точки зрения квантовых вычислений. Однако Китаев [25] описал семейство простых спиновых систем с локальными взаимодействиями, в которых возможно существование квазичастиц с неабелевыми фазами Берри. (Гамильтониан системы настолько фрустрирует спины, что основное состояние представляет собой чрезвычайно запутанное состояние с бесконечным радиусом квантовых корреляций.) Эти модели настолько просты (хотя, к сожалению, они требуют четырехчастичных взаимодействий), что можно даже представить такой материал, который достаточно хорошо описывается одной из моделей Китаева. Основные топологические свойства модели относительно безразличны к точным микроскопическим деталям, поэтому технологическая проблема «подгонки» параметров материала, возможно, не будет слишком сложной. Более того, если бы можно было управлять процессом переноса отдельных квазичастиц (возможно, при помощи подходящего магнитного пинцета), тогда эта система могла бы функционировать как отказоустойчивый квантовый компьютер.

Модель Китаева представляет собой систему спинов, располагающихся на ребрах квадратной решетки. Гамильтониан выражается в виде суммы двух взаимно коммутирующих четырехчастичных операторов, один из которых соответствует узлам решетки, а второй — плакетам (см. рис. 17). Поскольку они взаимно коммутируют, гамильтониан несложно диагонализовать путем диагонализации каждого его слагаемого по отдельности. Операторы на узлах напоминают локальные калибровочные симметрии (действующие независимо на каждом узле), а состояние, минимизирующее эти слагаемые, инвариантно относительно локальной симметрии, подобно физическим состояниям, подчиняющимся закону Гаусса в калибровочной теории. Операторы на плакетах подобны операторам «магнитного потока» в калибровочной теории, а эти слагаемые минимизируются, когда магнитный поток всюду обращается в нуль. Возбуждениями в такой системе являются состояния, в которых закон Гаусса нарушен на изолированных узлах (эти точки представляют собой «электрически заряженные» квазичастицы), и состояния, в которых магнитные потоки не равны нулю на изолированных плакетах (квазичастицы магнитного потока). Квантовое запутывание основного состояния таково, что нетривиальная фаза Берри, связанная с обходом заряда вокруг потока, идентична фазе Аароновa–Бома в аналогичной калибровочной теории.

Эти явления Аароновa–Бома стабильны даже при деформации гамильтониана данной модели. Действительно, если деформация достаточно мала, мы можем изучать ее влияние, используя теорию возмущений. Но пока воз-

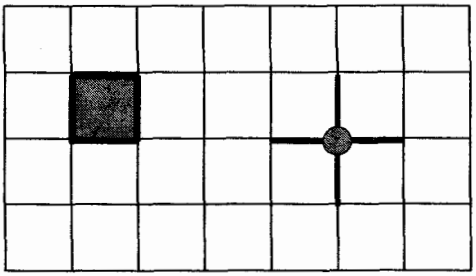


Рис. 17. Спиновая модель Китаева. Спины располагаются на ребрах решетки. Взаимодействуют четверки спинов, встречающихся на одном узле или окружающих один плакет

мущения пространственно локальны, топологические эффекты устойчивы, так как результаты теории возмущений определяются лишь суммой локализованных воздействий. Любая деформация модели, разрушающая дальнодействующие топологические взаимодействия, должна иметь непертурбативный характер.

Можно предвидеть два типа непертурбативных эффектов [58]. Основным состоянием теории мог бы оказаться «конденсат потока» с бесконечным количеством магнитных возбуждений. В таком случае возникло бы дальнодействующее взаимодействие притяжения между заряженными частицами и их античастицами. Разделение зарядов стало бы невозможным, и не возникло бы никаких дальнодействующих эффектов. В калибровочной теории это явление назвали бы *электрическим конфайнментом*. И наоборот, в основном состоянии мог бы образоваться конденсат электрически заряженных квазичастиц. Тогда возник бы конфайнмент магнитных возбуждений, и вновь дальнодействующие эффекты Аронова – Бома были бы разрушены. В калибровочной теории это назвали бы явлением Хиггса (или магнитным конфайнментом).

Таким образом, деформируя гамильтониан Китаева, мы можем ожидать, что в конечном счете столкнемся с границей раздела фаз, за пределами которой возникает электрический конфайнмент или явление Хиггса. Размер заключенной в эти границы области определяет, как именно должен быть изготовлен материал, чтобы он вел себя по предписанию Китаева. Чрезвычайно важный вопрос для разработчика материала состоит в следующем: смогут ли искусно подобранные *двухчастичные* взаимодействия так фрустрировать спиновую систему, чтобы в ней образовалось сильно запутанное

основное состояние, а между квазичастицами возбуждений возникли неабелевы взаимодействия Ааронова–Бома?

Дробный квантовый эффект Холла и модели Китаева преподнесли впечатляющий урок. Мы обнаружили калибровочные эффекты, возникающие как коллективные явления в системах с одними лишь короткодействующими взаимодействиями. Возможно, стоит подумать о том, что известные в природе калибровочные симметрии могут иметь подобное происхождение.

7.3. Топологические взаимодействия

Как уже отмечалось, в спиновых моделях Китаева существует два типа зарядов, переносимых локализованными квазичастицами, которые можно назвать «электрическими» и «магнитными» зарядами. В модели простейшего типа «магнитным потокам», переносимым частицами, можно сопоставить элементы некоторой конечной группы G, а «электрическим зарядам» — неприводимые представления[1] группы G. Если частица с зарядом, соответствующим неприводимому представлению $D^{(\nu)}$, квантовые числа которой закодированы во внутренней волновой функции $|\psi^{(\nu)}\rangle$, обходит вокруг потока, обозначенного групповым элементом $u \in G$, то ее волновая функция преобразуется по правилу

$$|\psi^{(\nu)}\rangle \to D^{(\nu)}(u)|\psi^{(\nu)}\rangle. \tag{37}$$

Используя это взаимодействие, мы можем *измерить* магнитный поток путем рассеяния на нем подходящей заряженной частицы [59]. Например, мы могли бы сконструировать изображенный на рисунке 18 флюксинтерферометр Маха–Цендера, чувствительный к относительной фазе, приобретаемой заряженными частицами, проходящими по траекториям справа или слева от потока. Если мы подходящим образом сбалансируем интерферометр, то сможем различить, скажем, два значения потока u_1, $u_2 \in G$; поток u_1 будет обнаруживаться появлением частицы в одном плече интерферометра, а поток u_2 — появлением частицы в другом плече. Конечно, сконструированный нами интерферометр не будет безупречным, но, тем не менее, измерение потока может быть отказоустойчивым: при наличии большого количества заряженных налетающих частиц и при многократном повторении измерения мы можем определить поток с достаточно высокой статистической достоверностью.

[1] Также возможно существование «дионов», переносящих заряды обоих типов; классификация переносимого дионом заряда достаточно тонкая, однако нам не потребуется детальное обсуждение свойств этих квазичастиц.

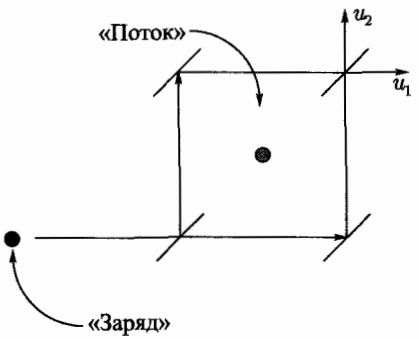


Рис. 18. Схематическое изображение интерферометра Маха–Цендера для измерения потока. Измеряемый поток помещается внутрь. Если поток имеет значение u_1, то пробный заряд появляется в одном плече, а при значении u_2 — в другом

Если два потока u_1 и u_2 принадлежат одному и тому же классу сопряженных элементов группы G, то существует связывающая их симметрия, а вся локальная физика не зависит от значения потока (см. ниже). Следовательно, когерентность суперпозиции потоков

$$a|u_1\rangle + b|u_2\rangle \tag{38}$$

не будет разрушаться локальными взаимодействиями с окружающей средой. Тем не менее, флюкс-интерферометр (действующий многократно) спроецирует флаксон на одно из двух собственных состояний $|u_1\rangle$ (с вероятностью $|a|^2$) или $|u_2\rangle$ (с вероятностью $|b|^2$).

Теперь представим, что два флаксона были тщательно откалиброваны, так что известно, что один из них несет поток u_1, а второй — поток u_2. Осторожно перемещая первый флаксон относительно второго, «переставим» их, как это показано на рисунке 19, после чего вновь выполним измерение потоков. После такого обмена, обход заряженной частицы вокруг правого флаксона топологически эквивалентен следующему обходу, совершаемому до перестановки: сначала вокруг правого флаксона, затем вокруг левого и, наконец, вокруг правого флаксона в противоположном направлении. Мы приходим к выводу, что обмен изменяет квантовые числа флаксонов по правилу

$$|u_1\rangle|u_2\rangle \to |u_2\rangle|u_2^{-1}u_1u_2\rangle, \tag{39}$$

что представляет собой нетривиальное взаимодействие, если два потока не коммутируют [60]. Таким образом, даже в отсутствие любых электрических

зарядов между некоммутирующими потоками существуют собственные интересные взаимодействия Аароно́ва–Бома. Так как обход одного потока вокруг другого может привести к *сопряжению* его значения, два флаксона, несущие на себе сопряженные потоки, должны рассматриваться как *неразличимые* частицы [61]. Перестановка двух таких объектов может изменить их внутренние квантовые числа; эти неразличимые в двумерии частицы, которые подчиняются экзотической неабелевой разновидности статистики, мы будем называть *неабелевыми анионами* [62].

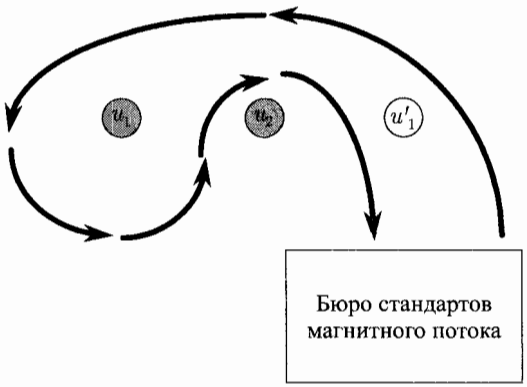


Рис. 19. Обменное взаимодействие потоков. Обозначенный u_1 поток из своего исходного положения (заштрихованное) перемещается в новое (не заштрихованное), а затем заново измеряется. Изображенная траектория заряженной частицы, что обходит исходное положение потока, топологически эквивалентна траектории, охватывающей новое положение; следовательно, значение потока u_1 меняется на $u'_1 = u_2^{-1} u_1 u_2$

Мы будем использовать обменное взаимодействие (39) в качестве фундаментальной логической операции в нашем квантовом компьютере Аароно́ва–Бома. Однако в действительности может оказаться удобным кодирование кубитов в парах флаксонов с тривиальным полным потоком [25]: мы будем рассматривать пары флаксон-антифлаксон вида $|u, u^{-1}\rangle$, но такие, в которых флаксон и антифлаксон удерживаются на достаточном расстоянии друг от друга, чтобы случайный обмен квантовыми числами между ними был маловероятен. Для выполнения логической схемы мы можем протащить одну пару сквозь другую, как это показано на рисунке 20. Так как полный поток, проходящий через середину внешней пары, тривиален, ее состояние не изменяется, но внутренние потоки сопрягаются внешним

потоком:

$$|u_1, u_1^{-1}\rangle|u_2, u_2^{-1}\rangle \to |u_2, u_2^{-1}\rangle|u_2^{-1}u_1u_2, u_2^{-1}u_1^{-1}u_2\rangle; \tag{40}$$

эта операция, очевидно, изоморфна результату перестановки отдельных потоков, описываемому уравнением (39). Использование пар вместо отдельных флаксонов имеет два преимущества. Во-первых, так как каждая пара имеет тривиальный полный поток, они не взаимодействуют, пока одна из них не протаскивается сквозь другую; следовательно, мы можем свободно перемещать пары по устройству, не вызывая никаких нежелательных взаимодействий с удаленными парами. Во-вторых, что более важно, пара может переносить заряд, даже если каждый ее элемент в этом отношении является нейтральным [63, 64]. Заряд пары можно измерить, и эта операция измерения заряда станет основным элементом универсального набора квантовых вентилей. Операцию (40) можно рассматривать как *классический* логический вентиль; она преобразует одно собственное состояние потока в другое. Чтобы выполнять интересные квантовые вычисления, мы должны уметь готовить когерентные суперпозиции собственных состояний потока. Именно это мы можем осуществлять, измеряя заряд пары.

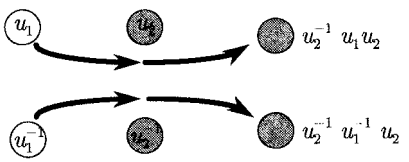


Рис. 20. Взаимодействие «протаскивания». Одна пара потоков протаскивается сквозь другую. Внешний поток не изменяется, но внутренний поток сопрягается внешним

Предположим, что u_0 и $u_1 \in G$ связаны соотношением $u_1 = v^{-1}u_0v$ для некоторой величины $v \in G$. Тогда, если мы рассматриваем собственные состояния потоков $|u_0, u_0^{-1}\rangle$ и $|u_1, u_1^{-1}\rangle$ в качестве состояний вычислительного базиса, эффект протаскивания одной из двух пар сквозь пару $|v, v^{-1}\rangle$ можно интерпретировать как NOT- или X-вентиль:

$$|u_0, u_0^{-1}\rangle \longleftrightarrow |u_1, u_1^{-1}\rangle \tag{41}$$

(см. рис. 21). Теперь предположим, что мы хотим приготовить одно из состояний

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|u_0, u_0^{-1}\rangle \pm |u_1, u_1^{-1}\rangle). \tag{42}$$

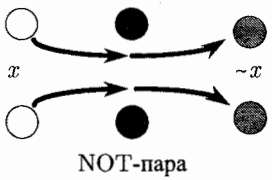


Рис. 21. NOT-вентиль. Протаскивание вычислительной пары потоков сквозь пару NOT обращает значение закодированного бита

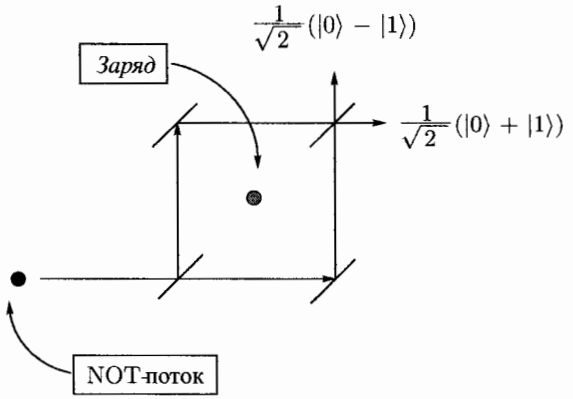


Рис. 22. Схематическое изображение интерферометра Маха – Цендера для измерения заряда. Пара потоков, заряд которой должен быть измерен, помещается внутрь. Если пробный поток NOT появляется в одном плече, то было приготовлено состояние заряда $|+\rangle$; если он появляется в другом плече, то было приготовлено состояние $|-\rangle$

Мы можем спроецировать когерентную суперпозицию $|u_0, u_0^{-1}\rangle$ и $|u_1, u_1^{-1}\rangle$ на базис $\{|\pm\rangle\}$, путем рассеяния флаксона $|v\rangle$ на паре или, дру-

гими словами, оперируя *зарядовым интерферометром*, как это показано на рисунке 22. Когда флаксон $|v\rangle$ движется вокруг пары, он приобретает тривиальную фазу Аронова–Бома, если пара находится в состоянии $|++\rangle$, и нетривиальную фазу -1, если пара находится в состоянии $|--\rangle$. Если интерферометр должным образом сбалансирован, то налетающая частица $|v\rangle$ будет детектироваться в одном плече интерферометра, когда состоянием пары является $|+\rangle$, и в другом, когда состоянием пары является $|-\rangle$. Это пример измерения заряда. Хотя интерферометр не будет безупречен, измерение заряда (как и измерение потока) может быть отказоустойчивым, если оно повторяется достаточное количество раз.

7.4. Универсальные топологические вычисления

Работая с парами флаксонов в качестве состояний вычислительного базиса, мы увидели, как выполняется операция перестановки (или «протаскивания») в (40), как измеряется поток (с использованием предварительно откалиброванных зарядов), и как измеряется заряд (с использованием предварительно откалиброванных потоков). Предположим теперь, что мы можем создать большой запас пар потоков. С помощью локальных процессов можно создать пары, несущие нулевой заряд и тривиальный поток; с точностью до нормировки, состояние такой пары имеет вид

$$|\text{charge zero}\rangle = \sum_u |u, u^{-1}\rangle, \tag{43}$$

где суммирование ведется по всему классу сопряженных элементов группы G. Поскольку при сопряжении любым элементом группы G это состояние остается неизменным, оно имеет тривиальные взаимодействия Аронова–Бома с любым потоком и, следовательно, не имеет детектируемого заряда. После рождения такой пары можно выполнить измерение потока, проецирующее ее состояние на одно из собственных состояний пары потоков $|u, u^{-1}\rangle$. Выполняя множество таких измерений для большого количества пар, мы собираем большой резервуар калиброванных пар потоков, которые при необходимости можно извлекать в процессе вычисления.

Но является ли наш квантовый компьютер универсальным — можем ли мы получить достаточное приближение любого желаемого унитарного преобразования? Чтобы исследовать этот вопрос, вспомним упомянутый в разделе 4.2 результат: для универсальных квантовых вычислений достаточно универсальных *классических* вычислений в совокупности с возможностью *выполнять* однокубитовые вентили X и Z и *измерять* X, Y и Z [17]. Действительно, существуют такие группы G, что операции (40) оказывается

достаточно для универсальных классических вычислений. Мы обнаружили [65], что если $G = A_5$ — группа четных перестановок пяти объектов, то из уравнения (40) можно построить вентиль Тоффоли. Например, в качестве состояний вычислительного базиса можно выбрать

$$u_0 = (125), \quad u_1 = (234); \tag{44}$$

то есть потоки, соответствующие циклам длины три (3-циклам) с одним общим объектом. Тогда вентиль Тоффоли можно построить в общей сложности из 16 элементарных операций «протаскивания»; кроме того, для ускорения выполнения этой операции используется шесть служебных пар. Ни в одной из меньших, чем A_5, групп вентиль Тоффоли не был обнаружен.[1] Так как A_5 также является наименьшей из конечных неразрешимых групп, напрашивается вывод, что неразрешимость — необходимое условие для порожденных сопряжением классических вычислений.[2]

Как уже говорилось, X-вентиль можно осуществить, протаскивая вычислительную пару вихрей сквозь пару с потоком v, таким, что $u_1 = v^{-1}u_0v$; здесь мы выбираем $v = (14)(35)$. Оказывается, Z-вентиль можно построить при помощи шести этапов «протаскивания» и четырех служебных пар. Измерение Z аналогично измерению потока, и мы уже видели, что измерение X можно выполнить путем измерения заряда пары, а именно, используя в зарядовом интерферометре в качестве налетающей частицы v. Остается лишь подтвердить, что мы можем измерить Y. Хотя по данной схеме измерение Y нельзя выполнить точно, оказывается, что вентиль «контролируемое Y» можно построить с помощью 31 этапа «протаскивания» и семи служебных пар. Обращаясь к другому изобретенному Китаевым трюку [67], для выполнения Y-измерения с любой желаемой точностью мы можем использовать вентиль «контролируемое Y» повторно.[3] Следовательно, мы построили набор универсальных вентилей, используя лишь взаимодействия Аароно́ва–Бома потоков и зарядов; мы располагаем отказоустойчивым универсальным квантовым компьютером.

[1] Ранее Китаев сообщал, что универсальные классические вычисления возможны для $G = S_5$.

[2] Конечная группа *неразрешима*, если она имеет нетривиальную подгруппу, коммутант которой совпадает с ней самой. (Коммутантом группы называется множество всех возможных произведений коммутаторов $aba^{-1}b^{-1}$ элементов рассматриваемой группы. Имеет место следующий критерий: если группа G некоммутативна и не имеет нетривиальных нормальных подгрупп, то она неразрешима. — *Прим. ред.*) Баррингтон [66] нашел признак разделения групп на разрешимые и неразрешимые по вычислительной сложности группового умножения.

[3] Фактически, измерение Y (который имеет собственные значения $\pm i$) с использованием вентиля контролируемое Y не работает, поскольку метод Китаева не различает связанные комплексным сопряжением собственные значения. То, что мы действительно построили — это вентиль контролируемое ωY, где $\omega = e^{2\pi i/3}$.

К сожалению, спиновая модель, на которой основана данная конструкция, не так проста. Так как группа A_5 имеет порядок 60, реализующая данную схему спиновая модель Китаева имеет 60-компонентный спин, расположенный на каждом (!) ребре решетки. Остается надеяться, что будет обнаружена более простая реализация вычислений Аронова–Бома.

7.5. Является ли природа отказоустойчивой?

Открытие коррекции квантовых ошибок и отказоустойчивости настолько изменило наше представление о квантовой информации, что впору задаться вопросом об их потенциальном значении для фундаментальной физики. Действительно, основные вопросы, имеющие отношение к потере квантовой информации, озадачивали физиков на продолжении двадцати лет.

В 1975 году, Стивен Хокинг [68] доказал, что квантовая информация неизбежно теряется при образовании черной дыры и ее последующем полном испарении. Суть доказательства предельно проста: из-за сильно искаженной причинной структуры пространства-времени черных дыр испускаемое излучение находится фактически на *том же* временном срезе, что и исчезающее за горизонтом событий коллапсирующее вещество. Если квантовая информация, первоначально закодированная в коллапсирующем веществе, должна в конечном счете возродиться в информации, закодированной в микросостоянии излучения, то она должна находиться в двух местах одновременно. Другими словами, квантовая информация должна быть *клонирована*, что, как известно, при обычных допущениях квантовой теории невозможно [69, 70]. Хокинг делает вывод, что не все физические процессы могут управляться унитарной временной эволюцией; законы квантовой теории нуждаются в пересмотре.

Эти аргументы убедительны, но многие физики им не доверяют. Возможно, одна из причин для скептицизма состоит том, что для природы выглядит странным допускать потерю даже маленького бита информации [71]. Если процессы с участием черных дыр могут разрушать информацию, то можно ожидать, что потеря информации неизбежна на масштабе порядка планковской длины $\left(G\hbar/c^3\right)^{1/2} \sim 10^{-33}$ см, на котором виртуальные черные дыры непрерывно возникают как квантовые флуктуации. Становится сложно понять, почему квантовая информация может быть так легко разрушена на планковском масштабе, но так хорошо сохраняется на больших расстояниях, которые мы можем исследовать экспериментально, — в конце концов, нарушения законов квантовой механики ни разу не наблюдались.

Наше сегодняшнее понимание отказоустойчивых квантовых вычислений дает полезный и потенциально продуктивный способ разобраться

в данной проблеме. В спиновых моделях Китаева мы можем представить, что разрушающие квантовую информацию локальные процессы достаточно просты. Тем не менее, если бы нам потребовалось проследить эволюцию системы с более грубым разрешением, отслеживая лишь информацию, закодированную в зарядах пространственно разделенных квазичастиц, мы с поразительной точностью наблюдали бы унитарную эволюцию; мы не обнаружили бы ни одного признака шума, происходящего ниже этого уровня.[1]

Таким образом, весьма приятно полагать, что Природа внесла в свою структуру отказоустойчивость, скрыв от нас квантовый шум на масштабе Планка. Открытие того, что квантовые системы можно стабилизировать с помощью соответствующих методов кодирования, заставляет нас задаться вопросом: является ли природа отказоустойчивой? Если да, то квантовая механика может доминировать (с превосходной точностью) на промежуточных масштабах длин, но спотыкаться как на планковском масштабе (где высока частота появления «ошибок»), так и на макроскопическом масштабе (где стремительна декогерентизация).

Настоящая работа была частично поддержана Управлением перспективного планирования оборонных научно-исследовательских работ (DARPA), грант DAAH04-96-1-0386, управляемым Военным исследовательским центром, и Министерством энергетики, грант DE-FG03-92-ER40701. Я благодарен полезным беседам и переписке с Доритом Аароновым, Дэвидом Бэкманом, Джоном Кортизом, Эриком Деннисом, Дэвидом Дивинченцо, Джара Эвслином, Крисом Фуксом, Шамом Какаде, Алексеем Китаевым, Мэнни Ниллом, Раймондом Лафламмом, Эндрю Ландалом, Сетом Ллойдом, Майклом Нильсеном, Уолтом Огберном, Питером Шором, Эндрю Стином и Кристофом Залкой. Я отдельно благодарю Дэниэла Готтесмана за множество плодотворных дискуссий об отказоустойчивых квантовых вычислениях.

Литература

[1] R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.*, **21**, 467–482 (1982); перевод: Р. Фейнман, Моделирование физики на компьютерах. *Квантовый компьютер и квантовые вычисления*. — Ижевск, РХД (1999).

[1] Аналогичный язык можно было бы использовать для характеристики осуществления каскадного кода: ошибки редки, когда мы исследуем квантовую информацию при слабом разрешении, но появляются намного чаще, если мы рассматриваем кодовый блок на более глубоких уровнях каскадирования.

[2] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, *Proc. Roy. Soc. Lond.*, A **400**, 97–117 (1985); перевод: Д. Дойч, Квантовая теория, принцип Черча–Тьюринга и универсальный квантовый компьютер. *Квантовый компьютер и квантовые вычисления.* — Ижевск, РХД (1999).

[3] P. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Fundamentals of Computer Science* (Los Alamitos, CA, IEEE Press, 1994), pp. 124–134; Расширенная версия: SIAM J. Comp. **26**, 1484–1509 (1997), online preprint quant-ph/9508027; перевод: П. Шор, Полиномиальные по времени алгоритмы разложения числа на простые множители и нахождения дискретного логарифма для квантовых компьютеров. *Квантовый компьютер и квантовые вычисления.* — Ижевск, РХД (1999).

[4] R. Landauer, Is quantum mechanics useful? *Phil. Tran. R. Soc. Lond.*, **353**, 367–376 (1995).

[5] R. Landauer, The physical nature of information, *Phys. Lett.*, A **217**, 188–193 (1996).

[6] R. Landauer, Is quantum mechanically coherent computation useful? In *Proc. Drexel-4 Symposium on Quantum Nonintegrability-Quantum-Classical Correspondence*, Philadelphia, PA, 8 September 1994, ed. D. H. Feng and B.-L. Hu (Boston, International Press, 1997).

[7] W.G. Unruh, Maintaining coherence in quantum computers, *Phys. Rev.*, A **51**, 992–997 (1995).

[8] S. Haroch and J.-M. Raimond, Quantum computing: dream or nightmare? *Phys. Today*, **49** (8), 51–52 (1996).

[9] W.H. Zurek, Decoherence and the transition from quantum to classical, *Phys. Today*, **44**, 36–44 (1991).

[10] P. Shor, Scheme for reducing decoherence in quantum computer memory, *Phys. Rev.*, A **52**, R2493–R2496 (1995).

[11] A.M. Steane, Error correcting codes in quantum theory, *Phys. Rev. Lett.*, **77**, 793–797 (1996).

[12] A.M. Steane, Multiparticle interference and quantum error correction, *Proc. Roy. Soc. Lond.*, A **452**, 2551–2577 (1996).

[13] J. von Neumann, Probabilistic logics and synthesis of reliable organisms from unreliable components, in *Automata Studies*, ed. C. E. Shannon and J. McCarthy (Princeton, Princeton University Press, 1956).

[14] P. Gács, Reliable computation with cellular automata, *J. Comp. Sys. Sci*, **32**, 15–78 (1986).

[15] P. Shor, Fault-tolerant quantum computation, in *Proceedings of the Symposium on the Foundations of Computer Science*, Los Alamitos, CA, IEEE Computer Society Press, 1996, pp. 56–65 (online preprint http://lanl.arxiv.org/abs/quant-ph/9605011, 1996).

[16] A.M. Steane, Active stabilization, quantum computation and quantum state synthesis, *Phys. Rev. Lett.*, **78**, 2252–2255 (1997).

[17] D. Gottesman, A theory of fault-tolerant quantum computation, *Phys. Rev.* A **57**, 127–137 (1998) (online preprint http://lanl.arxiv.org/abs/quant-ph/9702029, 1997).

[18] E. Knill and R. Laflamme, Concatenated quantum codes, Techn. Report LAOR-96-2808. (online preprint http://lanl.arxiv.org/abs/quant-ph/9608012, 1996).

[19] E. Knill, R. Laflamme, and W. Zurek, Accuracy threshold for quantum computation, (online preprint http://lanl.arxiv.org/abs/quant-ph/9610011, 1996).

[20] E. Knill, R. Laflamme, and W. Zurek, Resilient quantum computation: error models and thresholds, *Proc. Roy. Soc. Lond.* A **454**, 365–384 (1998) (online preprint http://lanl.arxiv.org/abs/quant-ph/9702058, 1997).

[21] D. Aharonov and M. Ben-Or, Fault tolerant quantum computation with constant error. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* ACM. New York, 1998, p.176. (online preprint http://lanl.arxiv.org/abs/quant-ph/9611025, 1996).

[22] А.Ю. Китаев, Квантовые вычисления: алгоритмы и исправления ошибок, *Успехи мат. наук*, **52**, стр. 53–112 (1997).

[23] J. Preskill, Reliable quantum computers, *Proc. Roy. Soc. London* A, **454**, pp. 385–410 (1998) (online preprint http://lanl.arxiv.org/abs/quant-ph/9705031, 1997); перевод: Дж. Прескилл, Надежные квантовые компьютеры (в этом издании).

[24] C. Zalka, Threshold estimate for fault tolerant quantum computing (online preprint http://lanl.arxiv.org/abs/quant-ph/9612028, 1996).

[25] A.Yu. Kitaev, Fault-tolerant quantum computation by anyons, *Annals of Phys*, **303**, 2–30 (2003), (online preprint http://lanl.arxiv.org/abs/quant-ph/9707021, 1997).

[26] F.J.MacWilliams, and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, (New York, North-Holland Publishing Company, 1977); перевод: Ф.Дж. Мак-Вильямс, Н.Дж. Слоэн, *Теория кодов, исправляющих ошибки*, Связь, М., 1979.

[27] E. Knill and R. Laflamme, A theory of quantum error-correcting codes, *Phys. Rev.* A **55**, 900–911 (1997).

[28] A.R. Calderbank and P.W. Shor, Good quantum error-correcting codes exist, *Phys. Rev.*, A **54**, 1098–1105 (1996).

[29] D. Gottesman, Class of quantum error-correcting codes saturating the quantum Hamming bound. *Phys. Rev.*, A **54**, 1862–1868 (1996).

[30] A.R. Calderbank, E.M. Rains, P.W. Shor, and N.J.A. Sloane, Quantum error correction and orthogonal geometry, *Phys. Rev. Lett.*, **78**, 405–408 (1997).

[31] A.R. Calderbank, E.M. Rains, P.W. Shor, and N.J.A. Sloane, Quantum error correction via codes over GF(4). IEEE Trans. Inf. Theory, 44(4), pp. 1369–1387 (online preprint http://lanl.arxiv.org/abs/quant-ph/9608006, 1996).

[32] J. Evslin, S. Kakade, and J. Preskill, unpublished (1996).

[33] D. DiVincenzo and P. Shor, Fault-tolerant error correction with efficient quantum codes. *Phys. Rev. Lett.*, **77**, 3260–3263 (1996).

[34] A.Yu. Kitaev, Quantum error correction with imperfect gates, in *Proceedings of the Third International Conference on Quantum Communication and Measurement*, Ed O. Hirota, A.S. Holevo, and C.M. Caves, pp 181–188 (New York, Plenum, 1997).

[35] D. Gottesman, Stabilizer codes and quantum error correction. Ph.D. thesis, California Institute of Technology (online preprint http://lanl.arxiv.org/abs/quant-ph/9705052, 1997).

[36] C. Bennett, D. DiVincenzo, J. Smolin, and W. Wootters, Mixed state entanglement and quantum error correction, *Phys. Rev.*, A **54**, 3824–3851 (1996).

[37] R. Laflamme, C. Miquel, J.P. Paz, and W. Zurek, Perfect quantum error correction code, *Phys. Rev. Lett.*, **77**, 198–201 (1996).

[38] D. Gottesman and J. Preskill, unpublished (1997).

[39] D. Gottesman, J. Evslin, S. Kakade, and J. Preskill, unpublished (1996).

[40] K. Obenland and A.M. Despain, Simulation of factoring on a quantum computer architecture, in *Proceedings of the 4th Workshop on Physics and Computation*, Boston, November 22–24, 1996, (Boston, New England Complex Systems Institute, 1996).

[41] K. Obenland, and A.M. Despain, Impact of errors on a quantum computer architecture, Technical Report, Information Science Institute, University of Southhern California, Oct 1, 1996; (online preprint http://www.isi.eu/acal/quantum/quantumoperrors.ps, 1996).

[42] C. Miquel, J.P. Paz, and W.H. Zurek, Quantum computation with phase drift errors, *Phys. Rev. Lett.*, **78**, 3971–3974 (1997); (online preprint http://lanl.arxiv.org/abs/quant-ph/9704003, 1997).

[43] J.I. Cirac and P. Zoller, Quantum computations with cold trapped ions, *Phys. Rev. Lett.*, **74**, 4091–4094 (1995).

[44] M.B. Plenio and P.L. Knight, Decoherence limits to quantum computation using trapped ions, *Proc. Roy. Soc. Lond.*, A **453**, 2017–2041 (1997).

[45] M. Grassl, Th. Beth, and T. Pellizzari, Codes for the quantum erasure channel, *Phys. Rev.*, A **56**, 33–38 (1997).

[46] A.K. Lenstra, J. Cowie, M. Elkenbracht–Huizing, W. Furmanski, P.L. Montgomery, D. Weber, J. Zayer, RSA factoring-by-web: the world-wide status (online document http://www.npac.syr.edu/factoring/status.html, 1996).

[47] D. Beckman, A. Chari, S. Devabhaktuni, and J. Preskill, Efficient networks for quantum factoring, *Phys. Rev.*, A **54**, 1034–1063 (1996).

[48] A.M. Steane, Space, time, parallelism and noise requirements for reliable quantum computing *Fortsch. Phys.*, **46**, 443–458 (1998); (online preprint http://lanl.arxiv.org/abs/quant-ph/9708021, 1997).

[49] C. Monroe, D.M. Meekhof, B.E. King, W.M. Itano, and D.J. Wineland, Demonstration of a fundamental quantum logic gate, *Phys. Rev. Lett.*, **75**, 4714–4717 (1995).

[50] Q.A. Turchette, C.J. Hood, W. Lange, H. Mabuchi, and H.J. Kimble, Measurement of conditional phase shifts for quantum logic. *Phys. Rev. Lett.*, **75**, 4710–4713 (1995).

[51] D.G. Cory, A.F. Fahmy, and T.F. Havel, Nuclear magnetic resonance spectroscopy: an experimentally accessible paradigm for quantum computing. *In Proceedings of the 4th Workshop on Physics and Computation*, T. Toffoli, M. Biafore, and J. Leao (eds.), Boston, New England Complex Systems Institute, pp. 87–91 (1996).

[52] N. Gershenfeld and I. Chuang, Bulk spin resonance quantum computation. *Science*, **275**, 350–356 (1997).

[53] J. Preskill, Quantum computing: pro and con *Proc. Roy. Soc. Lond.* A **454**, 469–486 (1998) (online preprint http://lanl.arxiv.org/abs/quant-ph/9705032, 1997); перевод в *Квантовые вычисления: за и против.* — РХД, Ижевск (1999).

[54] S. Lloyd, Universal quantum simulators, *Science*, **273**, 1073–1078 (1996); correction in *Science* **279**, 1113–1117 (1998).

[55] R. Prange and S. Girvin (eds.), *The Quantum Hall Effect*, (New York, Springer-Verlag, 1987); перевод *Квантовый эффект Холла*, под ред. Р. Пренджа и С. Гирвина. — М.: Мир (1989).

[56] N. Read and E. Rezayi, Quasiholes and fermionic zero modes of paired fraction quantum Hall states: the mechanism for nonabelian statistics, *Phys. Rev.* B, **54**, 16864–16887 (1996); (online preprint http://lanl.arxiv.org/abs/cond-mat/9609079, 1996).

[57] C. Nayak and F. Wilczek, $2n$ quasihole states realize 2^{n-1}-dimensional spinor braiding statistics in paired quantum Hall states, *Nucl. Phys.* B, **479**, 529–553 (1996); (online preprint http://lanl.arxiv.org/abs/cond-mat/9605145, 1996).

[58] G. t'Hooft, On the phase transition toward permanent quark confinement, *Nucl. Phys.*, B **138**, 1–25 (1978).

[59] M. Alford, S. Coleman, and J. March-Russell, Disentangling nonabelian discrete quantum hair, *Nucl. Phys.*, B **351**, 735–748 (1991).

[60] F.A. Bais, Flux metamorphosis, *Nucl. Phys.*, B **170**, 32–43 (1980).

[61] H.-K. Lo and J. Preskill, Nonabelian vortices and nonabelian statistics, *Phys. Rev.*, D **48**, 4821–4834 (1993)

[62] G. Moore and N. Read, Nonabelions in the fractional quantum Hall effect, *Nucl. Phys.*, B **360**, 362–396 (1991).

[63] M.G. Alford, K. Benson, S. Coleman, J. March-Russell, and F. Wilczek, Interactions and excitations of nonabelian vortices, *Phys. Rev. Lett.*, **64**, 1632–1635 (1990).

[64] J. Preskill and L.M. Krauss, Local discrete symmetry and quantum mechanical hair, *Nucl. Phys.*, B **341**, 50–100 (1990).

[65] R.W. Ogburn and H. Preskill, Topological quantum computation, in *Lect. Notes in Comp. Sci.* C.P. Williams (ed.) **1509**, 341–356, Springer–Verlag (1999).

[66] D.A. Barrington, Bounded width polynomial size branching programs recognoze exactly those languages in NC^1, *J. Comp. Sys. Sci.*, **38**, 150–164 (1989).

[67] A.Yu. Kitaev, Quantum measurements and the abelian stabilizer problem (online preprint http://lanl.arxiv.org/abs/quant-ph/9511026, 1995).

[68] S.W. Hawking, Breakdown of predictability in gravitational collapse, *Phys. Rev.*, D **14**, 2460–2473 (1976).

[69] D. Dieks, Communication by electron-paramagnetic-resonance devices. *Phys. Lett.*, A **92**, 271–272 (1982).

[70] W.K. Wootters and W.H. Zurek, A single quantum cannot be cloned, *Nature* **299**, 802–803 (1982); *Nature* **304**, 188-189 (1983).

[71] T. Banks, L. Susskind, and M.E. Peskin, Difficulties for the evolution of pure states into mixed states, *Nucl. Phys.*, B **244**, 125–134 (1984).

Интересующие Вас книги нашего издательства можно заказать почтой или электронной почтой:

subscribe@rcd.ru

Внимание: дешевле и быстрее всего книги можно приобрести через наш Интернет-магазин:

http://shop.rcd.ru

Книги также можно приобрести:

1. Москва, ИМАШ, ул. Бардина, д. 4, корп. 3, к. 415,
   тел.: (499) 135–54–37, (495) 641–69–38
2. МГУ им. Ломоносова (ГЗ, 1 этаж)
3. Магазины:

   Москва: «Дом научно-технической книги» (Ленинский пр., 40)

   «Московский дом книги» (ул. Новый Арбат, 8)

   Книжный магазин «ФИЗМАТКНИГА» (г. Долгопрудный, Новый корпус МФТИ, 1 этаж, тел. 409–93–28)

   С.-Пб.: «С.-Пб. дом книги» (Невский пр., 28)

Джон Прескилл

Квантовая информация и квантовые вычисления

Том 2

Дизайнер В. А. Толстолуцкая
Технический редактор А. В. Широбоков
Компьютерная верстка А. В. Моторин
Корректор О. З. Логунова

Подписано в печать 23.11.2011. Формат $60 \times 84^{1}/_{16}$.
Печать офсетная. Усл. печ. л. 18,14. Уч. изд. л. 19,45.
Гарнитура Таймс. Бумага офсетная № 1. Заказ № 11-55.

АНО «Ижевский институт компьютерных исследований»
426034, г. Ижевск, ул. Университетская, 1.
http://shop.rcd.ru E-mail: mail@rcd.ru Тел./факс: (+73412) 500–295

Дж. Прескилл — известный физик-теоретик, профессор теоретической физики отделения физики, математики и астрономии Калифорнийского технологического института (Калтех). Область научных интересов — физика элементарных частиц и космология, топологические дефекты, непертурбативные методы квантовой теории поля, квантовые аспекты ранней Вселенной и черных дыр. В середине 90-х годов увлекся теорией квантовой информации, квантовых вычислений и кодирования. В настоящее время — один из ведущих специалистов в этой области. Руководитель Института квантовых вычислений, а также Центра физики информации при Калтехе.

ISBN 978-5-4344-0030-5