Quantum Error Correction

Björn Stein

Abstract

Realisierungen von Quantengattern und -leitungen sind grundsätzlich fehlerbehaftet. Damit Quantencomputer skalieren, müssen die auftretenden Fehler korrigiert werden. Dabei treten einige Schwierigkeiten auf: Fehler können kontinuierlich sein (d.h. es gibt überabzählbar viele verschiedene Fehler), Berechungen können wegen des Non-Cloning-Theorems nicht redundant ausgeführt werden, und Messungen verändern den Systemzustand. Es wird gezeigt, dass Fehlerkorrektur dennoch nicht nur möglich ist, sondern es sogar erlaubt, beliebig lange Berechnungen auf einem fehlerbehafteten Quantencomputer mit logarithmischem Mehraufwand durchzuführen.

1 Schwierigkeiten der QEC

Realisierungen von Quantengattern und -leitungen sind grundsätzlich fehlerbehaftet: Der tatsächliche quantenmechanische Zustand weicht von dem gewünschten Zustand ab. Insbesondere scheint es unrealistisch, Fehlerwahrscheinlichkeiten wie bei klassischen Computern so sehr zu reduzieren, dass sie für praktische Rechnungen irrelevant sind, denn anders als ein klassischer Computer kann ein Quantencomputer seine Zustände nicht durch Verstärkung und Diskretisierung for Fehlern bewahren. Damit Quantencomputern skalieren, müssen die auftretenden Fehler korrigiert werden.
Dies schien zunächst (ca. 1985 bis '95) unmöglich, denn:

- **No-Cloning-Theorem** Quantale Zustände können nicht exakt dupliziert werden. **Erzeugung der zur Fehlerkorrektur nötigen Redundanz ist nicht trivial.**
- **Fehler sind kontinuierlich** Die Parameter a und b des Zustands $a\,|0\rangle + b\,|1\rangle$ können durch Fehler beliebig und kontinuierlich verändert werden. **Überabzählbar viele Fehler müssen korrigiert werden.**

- **Kollaps der Wellenfunktion** Jede Messung verändert den zugehörigen Teil der Wellenfunktion. Daher darf keine Messung durchgeführt werden, die Rückschlüsse auf die logischen QuBits erlaubt. **Feststellen eines Fehlers ist nicht trivial.**

2 Subsysteme

Um Fehlerkorrektur betreiben zu können, muss festgestellt werden, was für ein Fehler vorliegt. Dies muss nicht notwendigerweise mit einer Messung geschehen (siehe [4]). Dennoch ist es üblich und für die später behandelten fehlertoleranten Schaltungen sogar zweckmäßig, vorliegende Fehler per Messung zu bestimmen. Analog zum Fehlersyndrom in der klassischen Fehlerkorrektur spricht man dabei von einer **Syndrommessung**.
Eine Syndrommessung darf keine Informationen über den logischen (codierten) Zustand liefern, da dieser Zustand sonst in den dem Messergebnis entsprechenden Untervektorraum seines Hilbertsraumes kollabiert würde (projektive von-Neumann - Messung). Man bezeichnet Fehlersyndrom und logischen Zustand daher als unabhängige **Subsysteme**.
Ein triviales Beispiel für zwei Subsysteme eines 2-QuBit-Systems sind die einzelnen QuBits des Systems. Diese sind unabhängig, wenn das System sich in einem Produktzustand befindet.
Allgemein kann vor der Zerlegung in QuBit-Mengen ein Basiswechsel durchgeführt werden. Dies kann mit einem unitären Operator T erreicht werden.

2.1 Fehlervermeidung durch Wahl eines rauschfreien Subsystems

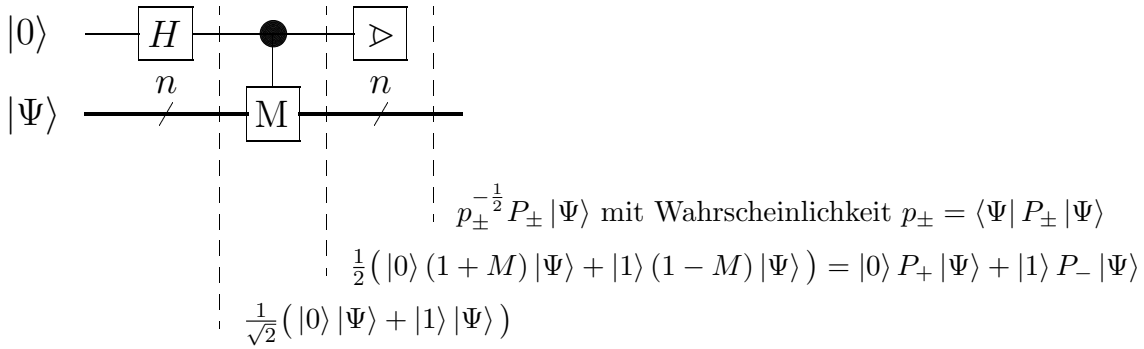
Bei der späteren theoretischen Betrachtung von Quantenfehlerkorrekturcodes wird es praktisch sein, Fehler, die auf verschiedene QuBits wirken, als statistisch unabhängig und gleichwahrscheinlich anzunehmen. Diese Annahme ist jedoch sehr künstlich und in den meisten Fällen nicht erfüllt.
Trotzdem ist diese Annahmen gerechtfertigt, da man einzelne Fehlerarten, von denen bekannt ist, dass sie auftreten, vollständig unterdrücken kann. Dies soll an folgendem Beispiel illustriert werden:
Basiswechsel durch Anwendung eines Operators T:

$$
\begin{aligned}
T\,|00\rangle &= |\mathbf{0}\rangle|\mathbf{0}\rangle \\
T\,|11\rangle &= |\mathbf{1}\rangle|\mathbf{0}\rangle \\
T\,|01\rangle &= |\mathbf{0}\rangle|\mathbf{1}\rangle \\
T\,|10\rangle &= |\mathbf{1}\rangle|\mathbf{1}\rangle
\end{aligned}
$$

Bei bekannten Hauptfehlertypen kann man durch Wahl eines geeigneten sog. **rauschfreien Subsystems** diese Fehler vermeiden.
Im Beispiel gibt es ein **rauschfreies Subsystem (rot)** bezüglich der Bit-Austausch-Fehler $|01\rangle \leftrightarrow |10\rangle$.
Physikalisches Beispiel: System mit Spinerhaltung
Übergänge von und nach $|\downarrow\downarrow\rangle$ und $|\uparrow\uparrow\rangle$ sind wegen Verletzung der Spinerhaltung verboten; Übergänge zwischen $|\uparrow\downarrow\rangle$ und $|\downarrow\uparrow\rangle$ treten jedoch auf.

2.2 Messung einer beliebigen Observablen

Ein Basiswechsel kann durch Transformation aller Operatoren (einschl. der Observablen M) erreicht werden. Wie aber misst man eine nicht-diagonale Observable M?
Eine Möglichkeit besteht darin, M zu diagonalisieren, also einen Basiswechsel explizit mittels Quantengattern durchführen. Es gibt aber auch eine elegantere Möglichkeit, nämlich folgenden Quanten-Algorithmus zur Messung einer Observablen $M = M^\dagger$:
Sei zur Analyse des Schaltbildes o.B.d.A. $M = |+\rangle\langle+| - |-\rangle\langle-| \equiv P_+ - P_-$ eine Messung mit den möglichen Ergebnissen $+1$ und -1.

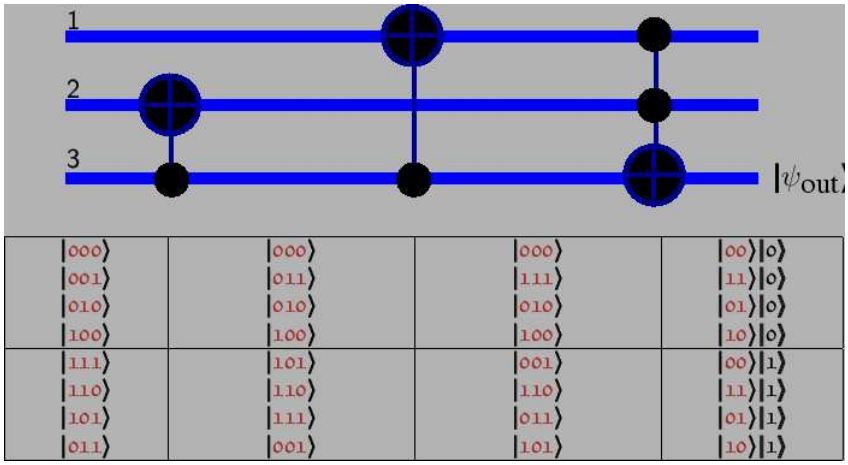


Die Rechnung zeigt, dass dieses Schaltbild äquivalent ist zur formalen Messung der Observablen M. Dies ist auch der Grund, warum gewöhnlich keine Quantenschaltelemente für die Messung spezieller Matrizen benutzt werden: Dieses Problem läßt sich auf die Messung einzelner QuBits in der kanonischen Basis zurückführen.

2.3 Repetition Code

Als zusätzliches, aber auch weiterführendes Beispiel für Subsysteme betrachte man den 3-QuBit-Wiederholungscode. Das logische, zu kodierende

QuBit sei $|\Psi_L\rangle = a\,|0_L\rangle + b\,|1_L\rangle$, während die tatsächlich verwendeten QuBits – also Codewörter und durch Fehler veränderte Codewörter – ohne Indizes geschrieben werden. Die Codewörter des 3-QuBit-Wiederholungscodes sind: $|0_L\rangle = |000\rangle$, $|1_L\rangle = |111\rangle$



Dargestellt ist eine kombinierte Korrektur- und Dekodierschaltung. Man beachte, dass die ersten zwei QuBits in den letzten beiden Spalten ein Fehlersyndrom darstellen, das angibt, welches QuBit einem sog. Bit–Flip– oder X–Fehler unterlag. Wie wir später sehen werden, gibt es für QuBits – anders als für klassische Bits – auch andere Fehlerarten, die dieser Code nicht korrigiert.
Häufig ist es wünschenswert, eine Korrektur, aber keine Dekodierung durchzuführen. Dazu ist es zweckmäßig, das Fehlersyndrom zu messen, und abhängig vom Messergebnis eine Operation durchzuführen, die alle QuBits in den richtigen Zustand bringt. Im Prinzip läßt sich aber auch dies ohne Messung und nur mit einem – dann deutlich komplizierteren – unitären Operator erreichen.[4]

2.4 Operatorsummendarstellung

Bevor wir Fehler korrigieren können, müssen wir erst Fehler beschreiben. Die allgemeinste vorstellbare (Fehler-) Operation $\mathcal{E}$ muss zwar der Quantenmechanik genügen, und damit durch einen unitären Operator U beschreibbar sein, kann aber nicht nur auf unseren Systemzustand $|\Psi_S\rangle$, sondern auch auf dessen Umwelt wirken. Da die Umgebung per Definition nicht Teil unseres Systems ist, ist diese Beschreibung jedoch häufig nicht hilfreich.

Sei o.B.d.A. die Umwelt im reinen Zustand $|0_E\rangle$. Dann gilt:

$$U\,|0_E\rangle\,|\Psi_S\rangle = \sum_k |k_E\rangle\,\langle k_E|\,U\,|0_E\rangle\,|\Psi_S\rangle$$

Auf das System wirken also die "environment labelled" Operatoren $U_k \equiv \langle k_E|\,U\,|0_E\rangle$
Die reduzierte Dichtematrix des Systems ρ wird durch $\mathcal{E}$ geändert in $\mathcal{E}(\rho) = \sum_k U_k \rho U_k^\dagger$.
Nomenklatur: $\mathcal{E} \equiv \{U_k\}$ ist die **Operatorsummendarstellung** für die durch den Operator U beschriebene Quantenoperation.
Die Operatorsummendarstellung ist nicht nur für die Formalisierung von Fehlern praktisch. Insbesondere kann auch die Fehlerkorrekturoperation in dieser Art beschrieben werden.
Eine mögliche Interpretation dieses Ergebnisses ist es, dass die Interaktion mit der Umgebung für das System eine Messung im Sinne einer projektiven von-Neumann-Messung bedeutet, denn innerhalb des Systems läßt sich dabei kein Unterschied zum betrachteten unitären Prozess feststellen. Aus gleicher Überlegung heraus kann man übrigens immer annehmen, dass in einem Quantenalgorithmus nicht mehr benutzte ("discarded") QuBits implizit gemessen werden. [4]

3 Fehler

3.1 1-QuBit-Fehler

Jeder mögliche 1-QuBit-Fehler F_1 ist durch eine unitäre 2×2 Matrix beschrieben, und kann als Linearkombination der folgenden formellen **1-QuBit-Fehlern** geschrieben werden:

$$F_1 = c_1\mathbf{1} + c_x X + c_y Y + c_z Z$$

$$\mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{"none"} \qquad X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{"Bit-Flip"}$$

$$Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{"Phase"} \qquad Y = -i\sigma_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad \text{"Bit-Phase"}$$

Man beachte, dass der kombinierte Bit-Phase–Fehler Y manchmal auch als identisch mit σ_y definiert wird (z.B. in [2]), was zu leichteren Kommutatorregeln führt. Uns werden jedoch folgende zwei Eigenschaften reichen:

$$AB = \pm BA \text{ für } A, B \in \{X, Y, Z\}; \quad Y = XZ$$

3.2 Beliebige Fehler

Allgemeiner **N-QuBit-Fehler**:

$$\sum_{ij\ldots}(c_{ij\ldots})E_i \otimes E_j \otimes \ldots$$

mit $E_i \in \{\mathbf{1}, X, Y, Z\}$, wobei in jedem Summanden max. N der $E_{\ldots}$ von $\mathbf{1}$ verschieden sind.
Ein QEC–Code schützt genau dann gegen beliebige N-QuBit-Fehler, wenn jeder formale N-QuBit-Fehler (jeder Summand in obiger Darstellung) unabhängig voneinander korrigiert wird.
Beweisidee:
Da ein Fehler ein linearer Operator ist, gilt $(\alpha A+\beta B)\left|\Psi\right\rangle = \alpha A\left|\Psi\right\rangle+\beta B\left|\Psi\right\rangle$ für beliebige Fehler A, B. Da die Korrekturoperation $\mathcal{R}$ die Fehler unabhängig voneinander korrigieren soll, und dies durch Anwendung linearer Operatoren erreichen kann, gilt dann auch $\mathcal{R}(\alpha A + \beta B)\left|\Psi\right\rangle = \alpha\mathcal{R}A\left|\Psi\right\rangle + \mathcal{R}\beta B\left|\Psi\right\rangle = \alpha\left|\Psi\right\rangle + \beta\left|\Psi\right\rangle = \left|\Psi\right\rangle$.
Die formalen N-QuBit-Fehler sind aufgrund der Konstruktion eine vollständige Basis für den Untervektorraum der N-QuBit-Fehler.

3.3 Versagen des 3-QuBit-Code

Der 3-QuBit-Repetition-Code von Seite 3 schützt vor 1-QuBit-X-Fehlern, nicht jedoch vor Z- oder Y-Fehlern; z. B.:

$$(1 \otimes 1 \otimes Z)(a\left|000\right\rangle + b\left|111\right\rangle) = a\left|000\right\rangle - b\left|111\right\rangle$$

Ein Z-Fehler kann also trotz Fehlerkorrektur aus $a\left|0_L\right\rangle + b\left|1_L\right\rangle$ den falschen Zustand $a\left|0_L\right\rangle - b\left|1_L\right\rangle$ machen!
Abhilfe:
Wegen $X = HZH$ kann man sich gegen Z- (aber nicht mehr gegen X-) Fehler schützen, indem man auf jedes der 3 QuBits eine Hadamard-Transformation H ausführt, einmal beim Codieren, und einmal beim Decodieren. Das ergibt einen Code, der aus folgenden Codewörtern besteht:

$$\begin{array}{lllll} \left|0_L\right\rangle & \propto & (\left|0\right\rangle + \left|1\right\rangle) & (\left|0\right\rangle + \left|1\right\rangle) & (\left|0\right\rangle + \left|1\right\rangle) \\ \left|1_L\right\rangle & \propto & (\left|0\right\rangle - \left|1\right\rangle) & (\left|0\right\rangle - \left|1\right\rangle) & (\left|0\right\rangle - \left|1\right\rangle) \end{array}$$

3.4 9–QuBit–Shor–Code

Idee: Vereinige Schutz vor X- und Z-Fehlern durch hierarchische QEC
Kodiere den logischen Zustand zunächst im 3-QuBit-Z-Code;
jedes dieser 3 QuBits anschließend im 3-QuBit-X-Code:

$$\begin{array}{lllll} |0_L\rangle & \propto & (|000\rangle + |111\rangle) & (|000\rangle + |111\rangle) & (|000\rangle + |111\rangle) \\ |1_L\rangle & \propto & (|000\rangle - |111\rangle) & (|000\rangle - |111\rangle) & (|000\rangle - |111\rangle) \end{array}$$

Dieser 9-QuBit-Code [5] schützt vor beliebigen 1-QuBit-Fehlern.
Beweis-Skizze:
Einzelne X- und Z-Fehler werden vom jeweiligen Code korrigiert.
Einzelne Y-Fehler wirken wie $Y = XZ$, und werden in diesen Teilschritten korrigiert.

3.5 Schranken

Um alle Fehler unabhängig voneinander korrigieren zu können, muss jedem ein eindeutiges, voneinander linear unabhängiges Syndrom zugeordnet werden.
Wenn in n QuBits k logische QuBits kodiert werden können, gibt es für 2^k logischen Zustände insgesamt 2^n Basisvektoren. Bei sog. orthogonalen Codes gibt es dann für jeden logischen Zustand einen Satz von 2^{n-k} Basisvektoren, die durch die Fehlerkorrektur diesem Zustand eindeutig zugeordnet werden.
Wenn beliebige $t = \lfloor d-1 \rfloor/2$–QuBit–Fehler korrigiert werden sollen (mit einem sog. [[n,k,d]]–Code), gibt es $\sum_{i=0}^{t} 3^i \binom{i}{n}$ verschiedene Fehlerarten, die korrigiert werden müssen.
Diese Überlegung führt zur sog. **Quantum–Hamming–Schranke** für orthogonale Codes:

$$2^{n-k} \geq \sum_{i=0}^{t} 3^i \binom{i}{n}$$

Es ist bisher jedoch nicht bekannt, ob diese Schranke allgemein gilt, oder nur für orthogonale Codes.[4]
Die (später erläuterte) Konstruktion von sog. CSS-Quantencodes aus klassischen Codes erlaubt es, die **Gilbert–Varshamov–Schranke** für CSS-Codes anzugeben: [6, 4]

$$\frac{k}{n} \geq 1 - 2H\left(\frac{2t}{n}\right)$$

Diese Schranke läßt sich noch verbessern, wenn nicht die Existenz eines CSS-Codes sondern nur die Existenz eines Stabilizer–Codes (später definiert) gefordert wird: [3, 4]

$$\frac{k}{n} \geq 1 - 2\frac{t\log_2 3}{n} - H\big(\frac{2t}{n}\big)$$

Zum Vergleich lautet die klassische Gilbert–Varshamov-Schranke

$$\frac{k}{n} \geq 1 - H\big(\frac{t}{n}\big)$$

Sowohl im klassischen als auch im quantalen Fall bezeichnet H dabei die binäre Shannon-Entropie

$$H(x) \equiv -x\log_2(x) - (1-x)\log_2(1-x)$$

Beispiele für bekannte Quantenfehlerkorrektur-Codes : [[5,1,3]], [[8,3,3]], [[21,6,5]], [[23,1,7]], [[127,29,15]], [[127,43,13]]

3.6 Satz 1: Quantenfehlerkorrekturbedingung

Sei $C = \{|0_L\rangle, |1_L\rangle, \ldots\}$ ein Quantencode, $P = |0_L\rangle\langle 0_L| + |1_L\rangle\langle 1_L| + \ldots$ der Projektor auf den Code, und $\mathcal{E} = \{E_i\}$ ein Fehleroperator.
Dann existiert eine Operation $\mathcal{R}$, die $\mathcal{E}$ auf C korrigiert gdw.

$$PE_i^\dagger E_j P = \alpha_{ij} P$$

für eine hermitische Matrix α erfüllt ist. [4]
Beweisskize dafür, dass die Quantenfehlerkorrekturbedingung hinreichend für die Existenz eines Codes ist:
1. Diagonalisiere α: $d = u^\dagger \alpha u$ diagonal, $\mathcal{E} = \{F_k \equiv \sum_i u_{ik} E_i\}$
$PF_k^\dagger F_l P = \sum_{ij} u_{ki}^\dagger U_{jl} P E_i^\dagger E_j P = d_{kl} P$, $F_k P = \sqrt{d_{kk}} U_k P$; U_k unitär.
$\Rightarrow$ F_k rotiert Code auf Untervektorraum mit $P_k \equiv U_k P U_k^\dagger = F_k P U_k^\dagger / \sqrt{d_{kk}}$

$$P_l P_k = P_l^\dagger P_k = U_l P F_l^\dagger F_k P U_k^\dagger / \sqrt{d_{ll} d_{kk}} = U_l U_k^\dagger d_{kl} / \sqrt{d_{ll} d_{kk}} = 0,\ l \neq k$$

2. Führe eine projektive (Syndrom-) Messungen mit den P_k durch. Die Messung liefert das Ergebnis k, und als Fehlerkorrektur wende $U_k^\dagger$ an.
Messung und Korrektur zusammen ist also beschrieben durch $\mathcal{R}(\rho) = \sum_k U_k^\dagger P_k^\dagger \rho P_k U_k$.
Für Code-Zustände $\rho = |\Psi_L\rangle\langle\Psi_L|$:

$$U_k^\dagger P_k^\dagger F_l |\Psi_L\rangle = U_k^\dagger U_k P F_k^\dagger F_l P |\Psi_L\rangle / \sqrt{d_{kk}} = \delta_{kl}\sqrt{d_{kk}} |\Psi_L\rangle$$

Damit: $(\mathcal{R} \circ \mathcal{E})(\rho) \propto \rho$ q.e.d.

4 Stabilizer - Formalismus

Unter der Matrixmultiplikation bilden die formalen Fehleroperatoren auf n QuBits, multipliziert mit $\pm 1, \pm i$, die sog. Pauli-Gruppe G_n. Beispiel: $G_1 \equiv \{\pm \mathbf{1}, \pm i\mathbf{1}, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}$
Sei S eine Untergruppe von G. Dann gibt es einen Untervektorraum V_s des n-QuBit-Hilbertraums mit $|\Psi\rangle \in V_s \Leftrightarrow s\,|\Psi\rangle = |\Psi\rangle\, \forall s_l \in S$
Bezeichnung: S ist **Stabilisator** von V_s; V_s wird **stabilisiert** von S.
$S = \{s_l\} = \langle g_i \rangle$ ist bestimmt durch seine **Generatoren** g_i, wobei sich alle s_l als Produkte von Generatoren schreiben lassen.
Wichtige Eigenschaft:

$$-\mathbf{1} \notin S = \langle g_1, .., g_{n-k} \rangle \text{ mit } [g_i, g_j] = 0, \text{ unabh. } g_i \in G_n \Rightarrow \dim(V_s) = k; g_i = g_i^\dagger$$

$$dim(V_s) = 1 \Rightarrow \text{Zustand } |\Psi\rangle \text{ ist bis auf globale Phase } e^{i\Theta} \text{ festgelegt.}$$

4.1 Operatoren im Stabilizer–Formalismus

Die unitären Matrizen U mit $UG_nU^\dagger = G_n$ bilden den **Normalizer** von G_n.
Solche U werden vollständig beschrieben durch Wirkung auf den $S = \langle g_i \rangle$:

$$U\,|\Psi\rangle = Ug_i\,|\Psi\rangle = Ug_iU^\dagger U\,|\Psi\rangle$$

Der neue Zustand wird also stabilisiert von $USU^\dagger = \langle Ug_iU^\dagger \rangle$.

Obige Bedingung an U wird erfüllt von: CNOT, H, X, Y, Z, $S = \sqrt{Z}$
Nicht erfüllt wird sie z.B. von: $\pi/8$, Toffoli

Operator	Eingang	Ausgang
U	$Y = XZ$	$UYU^\dagger = UXU^\dagger UZU^\dagger$
CNOT	$X \otimes \mathbf{1}$	$X \otimes X$
	$\mathbf{1} \otimes X$	$\mathbf{1} \otimes X$
	$Z \otimes \mathbf{1}$	$Z \otimes \mathbf{1}$
	$\mathbf{1} \otimes Z$	$Z \otimes Z$
H	X	Z
	Z	X
$S = \sqrt{Z}$	X	Y
	Z	Z
X	X	X
	Z	$-Z$
Y	X	$-X$
	Z	$-Z$

4.2 Messungen im Stabilizer-Formalismus

Im System mit Zustand $|\Psi\rangle$ und Stabilisator $S = \langle g_1, .., g_n\rangle$ werde eine Observable $O \in G_n$ gemessen (d.h. $O = O^\dagger$). Mögliche Messergebnisse sind die Eigenwerte von O: ± 1

- Falls $Og_i = g_i O \ \forall i$: $g_i O |\Psi\rangle = O |\Psi\rangle \in V_S$. Wegen $O^2 = \mathbf{1}$: Entweder $+O \in S$ oder $-O \in S$. $|\Psi\rangle$ und S bleiben also unverändert, und die Messung liefert mit Wahrscheinlichkeit 1 das Ergebnis ± 1.

- Sonst: O.B.d.A. sei $Og_1 = -g_1 O$ und $Og_i = g_i O \ \forall i > 1$
  (falls $Og_i = -g_i O$ ersetze: $g_i \to g_i' \equiv g_1 g_i \Rightarrow Og_i' = g_i' 0$).

  Messergebnis ± 1 mit Projektor $P_\pm = (\mathbf{1} \pm O)/2$ und Wahrscheinlichkeit $p_\pm = \mathrm{tr}(P_\pm |\Psi\rangle \langle\Psi|) = \mathrm{tr}(P_\pm g_1 |\Psi\rangle \langle\Psi|) = \mathrm{tr}(g_1 P_\mp |\Psi\rangle \langle\Psi|) = \mathrm{tr}(P_\mp |\Psi\rangle \langle\Psi| g_1^\dagger) = p_\mp \Rightarrow p_\pm = 1/2$.
  Neuer Zust. $\sqrt{2} P_\pm |\Psi\rangle$ hat $S' = \langle \pm O, g_2, .., g_n\rangle$, da $[P_\pm, \pm O] = 0$

4.3 Gottesman-Knill Theorem

Ein Quanten-Algorithmus, der nur folgende Elemente benötigt, läßt sich in polynomialer Zeit auf einem klassischen Computer simulieren:

- QuBit - Präparation in der kanonischen Basis
- Beliebige Kombinationen der Gatter CNOT, H, X, Y, Z, S
- Messung von Observablen der Pauli-Gruppe G_n

- Klassische Kontrolle aufgrund der Messergebnisse

Beweisidee:
Alle diese Rechenelemente können im Stabilizer-Formalismus ausgedrückt werden, wobei für einen n-QuBits erfordernden Algorithmus nur mit den $O(n)$ Generatoren gearbeitet wird.
Ergebnis: $O(n^2 m)$ für m simulierte Elemente
$\Rightarrow$ Verschränkung reicht nicht für "mächtigen" Quantencomputer!

4.4 Stabilizer–Codes

V_S, stabilisiert durch eine Untergruppe $S = \langle s_i \rangle$ der Pauli-Gruppe G_n mit $-\mathbf{1} \notin S$ und $n-k$ unabhängigen Generatoren s_i, ist ein $[n,k]$ **Stabilizer-Code** $C(S)$.
Die Code-Wörter sind die Basisvektoren von V_S.
Weitere Definition:
Centralizer $Z(S) \equiv \{E \in G_n \| Eg = gE \ \forall g \in S\}$.
Ohne Beweis: Für $-\mathbf{1} \notin S$: $Z(S) = N(S) \equiv \left\{E \in G_n \| EgE^\dagger \in S \ \forall g \in S\right\}$

4.5 Satz 2: Fehlerkorrekturbedingung für Stabilizer–Codes

Sei $C(S)$ ein Stabilizer-Code. Dann ist $\{E_i\} \subset G_n$ eine Menge von korrigierbaren Fehlern, wenn $E_i^\dagger E_k \notin Z(S) - S = N(S) - S$.
Beweis:

- $E_i^\dagger E_k \in S \Rightarrow P E_i^\dagger E_k P = P = P E_k^\dagger E_i P$
- $E_i^\dagger E_k \in G_n - N(S) \Rightarrow \exists g_1 \in S : \left\{g_1, E_i^\dagger E_k\right\} = 0$.
  Wähle Generatoren g_j mit $\langle g_1, .., g_{n-k'} \rangle = S$ und $\left[g_j, E_i^\dagger E_k\right] = 0$

$$\Rightarrow P \propto \prod_{l=1}^{n-k} (\mathbf{1} + g_l) \Rightarrow E_i^\dagger E_k P \propto (\mathbf{1} - g_1) E_i^\dagger E_k \prod_{l=2}^{n-k} (\mathbf{1} + g_l)$$

Aus $(\mathbf{1} + g_1)(\mathbf{1} - g_1) = 0$ folgt $P(\mathbf{1} - g_1) = 0$ und $P E_i^\dagger E_k P = 0$.

$\Rightarrow P E_i^\dagger E_k P = \alpha_{ik} P = \alpha_{ki}^\star P \Rightarrow$ Nach Satz 1 ist $\{E_i\}$ korrigierbar.

5 Calderbank-Shor-Steane–Codes

Seien C_X ein $[n, k_1]$ und $C_Z^\perp \subset C_X$ ein $[n, k_2]$ klassischer linearer Code; C_X und C_Z korrigieren je t Fehler.
Definiere Quantencode $\mathbf{CSS}(C_X, C_Z^\perp)$ mit Codewörtern

$$\left| x \oplus C_Z^\perp \right\rangle \equiv \frac{1}{\sqrt{\left|C_Z^\perp\right|}} \sum_{y \in C_Z^\perp} |x \oplus y\rangle \,, \quad x \in C_X$$

Dabei bezeichne $\oplus$ die bit-weise Addition ohne Übertrag.
Es gibt $\frac{|C_X|}{|C_Z^\perp|} = 2^{k_1 - k_2}$ Codewörter
$\Rightarrow$ CSS$(C_X, C_Z^\perp)$ ist ein $[[n, k_2 - k_1]]$ Quantencode. [6, 4, 3]
Beweis, dass CSS t-QuBit-Fehler E korrigiert:

$$E = \bigotimes_{i=1}^{n} E_i = \bigotimes_{i=1}^{n} X^{e_{xi}} Z^{e_{zi}} \qquad \text{mit } n\text{−Bit − Vektoren } e_x, e_z$$

$E \left| x \oplus C_Z^\perp \right\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{y \in C_Z^\perp} (-1)^{(x \oplus y) \cdot e_z} |x \oplus y \oplus e_x\rangle$
Benutze Hilfs-QuBits $|0\rangle$ und den unitäre Operator U: $U |0\rangle |x \oplus y \oplus e_x\rangle \equiv |H_X(x \oplus y \oplus e_x)\rangle |x \oplus y \oplus e_x\rangle$, wobei H_X die Prüfmatrix von C_X ist, also: $H_X(x \oplus y \oplus e_x) = H_X e_x$.
Nach Anwendung von U liefert eine Messung der Hilfsbits daher $H_X e_x$, wodurch der Fehler e_x eindeutig bestimmt ist, da C_X diesen Fehler korrigiert.
Anwendung von $E_X^{-1} \equiv \bigotimes_{i=1}^{n} X^{e_{xi}}$ liefert:

$$E_X^{-1} \langle H_X e_x | \, U \, |0\rangle \, E \left| x \oplus C_Z^\perp \right\rangle = \frac{1}{\sqrt{\left|C_Z^\perp\right|}} \sum_{y \in C_Z^\perp} (-1)^{(x \oplus y) \cdot e_z} |x \oplus y\rangle$$

Jetzt sind X-Fehler bereits korrigiert, Z-Fehler nicht. Diese können jedoch entsprechend behandelt werden, indem sie mittels der Walsh-Hadamard-Operation H in X-Fehler transformiert werden: $HZH = X$
Nach bit-weiser Anwendung von H haben wir den Zustand

$$\frac{1}{\sqrt{2^n \left|C_Z^\perp\right|}} \sum_{z'} \sum_{y \in C_Z^\perp} (-1)^{(x \oplus y) \cdot (e_z \oplus z')} \left| z' \right\rangle = \sqrt{\frac{2^n}{|C_Z^\perp|}} \sum_{z \in C_Z} (-1)^{x \cdot z} |z \oplus e_z\rangle$$

Nach Korrektur des nunmehr als X-Fehler wirkenden e_z mittels C_Z und bit-weiser Anwendung von $H^{-1} = H$ ergibt sich der Zustand von vor der ersten

H-Anwendung, mit $e_z = \vec{0}$ gesetzt:

$$\frac{1}{\sqrt{|C_Z^\perp|}} \sum_{y \in C_Z^\perp} (-1)^{(x \oplus y)\cdot \vec{0}} |x \oplus y\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{y \in C_Z^\perp} |x \oplus y\rangle \quad \text{q.e.d.}$$

5.1 7-QuBit–"Steane"–Code

Der CSS$(C, C^\perp)$ Code mit $C = [7, 4, 3]$ Hamming-Code heisst aus historischen Gründen (7-QuBit-) Steane-Code.[6] Da $C^\perp$ und C 1-Bit-Fehler korrigieren, korrigiert der Steane-Code 1-QuBit-Fehler.

$$\begin{aligned} |0_L\rangle = 8^{-1/2}\big(& |0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\ + & |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle \big) \\ |1_L\rangle = 8^{-1/2}\big(& |1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\ + & |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle \big) \end{aligned}$$

Die Generatoren der Stabilisatoren für den Steane-Code sind:

$$\begin{array}{ccc} \mathbf{111}XXXX & \mathbf{1}XX\mathbf{11}XX & X\mathbf{1}X\mathbf{1}X\mathbf{1}X \\ \mathbf{111}ZZZZ & \mathbf{1}ZZ\mathbf{11}ZZ & Z\mathbf{1}Z\mathbf{1}Z\mathbf{1}Z \end{array}$$

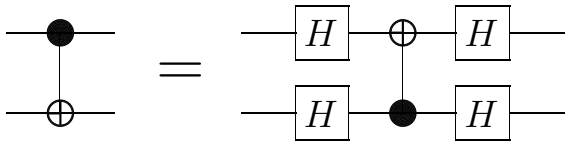
6 Fehlertolerantes Quantum Computing

Bis jetzt wurde nur die Korrektur einmal aufgetretener Fehler — also Fehler in den QuBit–Leitungen — mittels fehlerfreien Quantengattern betrachtet. Will man die Quantenfehlerkorrektur jedoch dazu einsetzen, den Einfluss vorhandener Gatterfehler in Implementationen von Quantencomputern zu verringern, so ist dieses Modell inkonsistent.
Eine Abschätzung der Effizienz der Quantenfehlerkorrektur bei Verwendung fehlerbehafteter Gatter ist schwierig:

- **Fehlertolerante** Schaltungen müssen gefunden werden, da die Fehlerfortpflanzung aus einem 1-QuBit-Fehler sonst 2-QuBit-Fehler (usw.) machen kann, die dann evtl. nicht mehr korrigierbar sind.
- **Der Aufwand** für lange Codes – unter Berücksichtigung der Fehlerfortpflanzung – ist schwer zu bestimmen (auch asymptotisch).

Abschätzung[7]: [[127,29,15]] Code erlaubt "large computations" bei Gatterfehlern von maximal 10^{-5}.

6.1 Beispiel für Fehlerfortpflanzung



Es ist klar, dass ein Bit-Flip-Fehler auf dem ersten ("Control-") QuBit ($X \otimes \mathbf{1}$) sich auf das zweite QuBit ausbreiten kann.
Doch auch in Gegenrichtung ist Fehlerfortpflanzung möglich:
Am Ersatzschaltbild wird ersichtlich, dass ein Phase-Flip - Fehler auf dem zweiten QuBit ($\mathbf{1} \otimes Z = \mathbf{1} \otimes HXH$) sich auf das erste ("Control-") QuBit ausbreiten kann!

6.2 Einfache Abschätzung

Die Schwierigkeit, festzustellen, wie Quantenfehlerkorrekturcodes sich asymptotisch für große Zahlen von QuBits und zu korrigierenden Fehlern verhalten unter der nicht-trivialen Einschränkung, nur fehlertolerante Schaltungen verwenden zu können, läßt durch die Konstruktion von hierarchisch verschachtelten Codes vermeiden. Diese Idee führt zu folgender Abschätzung: [1]

1-QuBit Fehlerwahrscheinlichkeit eines Gatters	$= p < C$
Fehlerw-keit mit L Verschachtelungsebenen	$\propto C^{-1}(Cp)^{2^L}$
Benötigte Ebenen L für T Gatteroperationen	$\propto \ln\ \ln\ T$
Overhead (Qubits, Gatter)	$\propto n^L \propto \ln T$

Dieses Schema erlaubt also, einen Algorithmus mit theoretischer Komplexität $O(T)$ auf einem realen Quantencomputer in $O(T \ln T)$ auszuführen.

7 Zusammenfassung

Quantenfehlerkorrektur ist möglich, muss aber mehr leisten als klassische Fehlerkorrektur, da mehr Fehler möglich sind. Daher gibt es für die Quantenfehlerkorrektur auch ungünstigere Schranken an die minimale Anzahl der zur Codierung nötigen QuBits und die maximale Anzahl der codierten QuBits und korrigierbaren Fehler.
Aus klassichen Fehlerkorrekturcodes lassen sich mit dem Verfahren von Calderbank, Shor und Steane Quantenfehlerkorrekturcodes konstruieren, die

jedoch nicht optimal sind.
Falls die Wahrscheinlichkeit für einen Fehler pro Quantengatter eine bestimmte Schranke — zwischen 10^{-3} und 10^{-6}, je nach genauen Annahmen — unterschreitet, so kann mittels Quantenfehlerkorrektur und fehlertolerantem Schaltungsdesign erreicht werden, dass beliebig genaue und damit beliebig lange Rechnungen ausgeführt werden können.

References

[1] D. Gottesman. Threshold for fault-tolerant computation, 1997. http://perimeterinstitute.ca/people/researchers/dgottesman/threshold.html.

[2] D. Gottesman. Group theory and quantum error correction, 2000. http://newton.kias.re.kr/∼jaewan/WS/Pres/Gottesman_1.pdf.

[3] J. Gruska. *Quantum Computing*. McGraw-Hill, London, 1999.

[4] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[5] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, 1995.

[6] A. M. Steane. Multiple particle interference and quantum error correction. *Proc. Roy. Soc. Lond. A*, 452:2551ff, 1996. quant-ph/9601029v3.

[7] A. M. Steane. Efficient fault tolerant quantum computing. *Nature*, 399:124–126, 1999. quant-ph/9809054.