Quantenfehlerkorrektur

Sebastian Smerat

14. Juli 2005

Zusammenfassung

Bis 1995 war man davon überzeugt, dass Quantencomputer in unserer Zukunft keine sinnvollen Aufgaben erfüllen können. Die Quantencomputer sind sehr fehleranfällig und die Korrektur von Fehlern wurde als unmöglich erachtet. Wie sollte man einen Zustand korrigieren, der bei seiner Messung zerstört werden würde? 1995 fand P.W.Shor eine Lösung für dieses Problem.
In diesem Artikel soll das Problem der Quantenfehler besprochen werden, nach einer Einleitung über Fehler bei klassischen Bits. Dazu gehört eine sinnvolle Darstellung der Fehler im Formalismus der Quantenmechanik und Methoden der Fehlerkorrektur.

1 Klassische Fehler

1.1 Bit-Flip

Der einzige Fehler, der bei klassischen Bits (kurz: cbit[1]) auftreten kann, ist der Bit-flip Fehler. Durch äußere Einwirkungen, zum Beispiel magnetische oder elektrische Felder kann der klassische Zustand eines cbits umklappen, ohne dass dies durch den Rechenprozess so gewollt war. Wird keine Fehlerkorrektur verwendet, so wird sich das Resultat der Rechnung als falsch herausstellen.
Für klassische Computer ist die Fehlerkorrektur jedoch verhältnismäßig einfach. Insbesondere durch die Tatsache, dass man cbits messen kann, ohne deren Zustand zu verändern.

[1] classical bit

1.2 Fehlerkorrektur durch Wiederholung

Betrachtet wird ein cbit $a = 0, 1$. Angenommen a sei im Zustand 0. Ein Bit-flip würde dann den Übergang

$$0 \to 1$$

verursachen.
Eine einfache Methode zur Fehlerkorrektur bietet die folgende Kodierung der cbits:

$$\bar{0} = 000$$
$$\bar{1} = 111$$

Jedes cbit wird also durch drei cbits ersetzt. Wieder das Beispiel a im Zustand 0. Kodierung bewirkt

$$a = \bar{0} \to 000$$

Ein einzelner Bit-flip erzeugt zum Beispiel den Fehlerzustand

$$000 \to 010$$

Da nun zwei der cbits im Zustand 0 sind, wird angenommen, dass a ursprünglich im Zustand $\bar{0}$ war und der Zustand $\bar{0}$ wird wiederhergestellt. Die Korrektur funktioniert nicht, wenn zwei Bit-flips auftreten. Dann würde zum Beispiel aus $a = 101$ der korrigierte Zustand $\bar{1}$ werden, obwohl a ursprünglich im Zustand $\bar{0}$ war.

1.2.1 Wie gut ist der Code?

Betrachtet wird zunächst ein Beispiel: Die Wahrscheinlichkeit, dass ein Bit-flip auftritt sei nun $p = 0.25$. Wenn ein zweifacher Fehler oder dreifacher Bit-flip-Fehler gemacht wird, kann dieser nicht korrigiert werden. Berechne nun die Wahrscheinlichkeit, dass zwei Bitflips auftreten. Beachte dabei, dass die Wahrscheinlichkeit, dass ein Bit nicht geflippt wird $1 - p = 0.75$ ist. Der Anfangszustand sei $\bar{0}$. Dann gilt für die Wahrscheinlichkeiten

$$p_{\bar{0}}(011) = 0.75 \cdot 0.25 \cdot 0.25 = 0.046875$$
$$p_{\bar{0}}(101) = 0.25 \cdot 0.75 \cdot 0.25 = 0.046875$$
$$p_{\bar{0}}(110) = 0.25 \cdot 0.25 \cdot 0.75 = 0.046875$$

Der Fehler wird natürlich auch nicht richtig korrigiert, falls 3 Bit-flips auftreten. Die Wahrscheinlichkeit dafür ist

$$p_{\bar{0}}(111) = (0.25)^3 = 0.015625$$

Also ergibt sich für die Wahrscheinlichkeit eines nicht korrigierbaren Fehlers

$$\begin{aligned} \bar{p} &= p_{\bar{0}}(011) + p_{\bar{0}}(101) + p_{\bar{0}}(110) + p_{\bar{0}}(111) \\ &= 0.15625 \end{aligned}$$

Dies ist eine deutliche Verbesserung gegenüber $p = 0.25$. Ohne diese Kodierung ist die Wahrscheinlichkeit eines Fehlers $p = 0.25$ und mit dieser Kodierung $p = 0.15625$.

Allgemein gilt für den 3-Bit-Wiederholungs-Code:

$$\bar{p} = 3 \cdot (1 - p)p^2 + p^3 \tag{1}$$

1.2.2 Das Versagen des Codes

Von grossem Interesse ist es zu wissen, bis zu welcher Wahrscheinlichkeit p eines Bit-flips die Fehlerkorrektur funktioniert. Berechne also den Schwellenwert p, ab dem also $\bar{p} \geq p$. Ab diesem Punkt wird die Fehlerkorrektur absolut nutzlos.
Benutze für $\bar{p}$ die Gleichung (1)

$$\begin{aligned} & 3 \cdot (1 - p)p^2 + p^3 \geq p \\ \Rightarrow \quad & 3 \cdot (1 - p)p + p^2 \geq 1 \\ \Rightarrow \quad & (p - \tfrac{3}{4})^2 \leq \tfrac{1}{16} \\ \Rightarrow \quad & \tfrac{1}{2} \leq p \leq 1 \end{aligned}$$

Das heißt also, dass für Wahrscheinlichkeiten $\frac{1}{2} \leq p \leq 1$ der Korrekturcode mehr Fehler erzeugt, als dies ohne Korrekturcode der Fall wäre.
Beispiel: Sei $p = 0.70$. Dann ist $\bar{p} = 0.784$. Also ist die Wahrscheinlichkeit eines Bit-Flips mit der 3-Bit-Wiederholung größer als ohne jegliche Kodierung.

1.3 Lineare Fehlerkorrekturcodes

Eine interessante Möglichkeit der Fehlererkennung sind die **linearen Codes**. Nun soll eine kurze Einführung in die Codierungstheorie, in lineare Codes und als Beispiel der Hamming-Code folgen.

1.3.1 Codierungstheorie

Unter einer **Codierung** versteht man eine injektive Abbildung $c : A \to B$ zwischen zwei Wortmengen A und B. Die Menge aller Codewörter, also das Bild $c(A)$ nennt man den **Code**.
Dies bedeutet aber noch nicht, dass jeder Code auch eindeutig umkehrbar ist, also eine Dekodierung sinnvoll durchführbar ist. Dazu muss die sogenannte **FANO-Bedingung** erfüllt sein:

> *Wenn kein Codewort Anfangswort eines anderen Codewortes ist, dann ist jede Zeichenkette $c(A) \subseteq B$ eindeutig dekodierbar.*

Ein Beispiel für einen nicht eindeutig umkehrbaren Code ist der *Morse-Code* ohne Pause:

$$c(usa) = c(idea) = \cdot\cdot - \cdot\cdot\cdot -$$

Für **Blockcodes** ist die FANO-Bedingung immer erfüllt.

Definition 1.1 *Sei $n \in \mathbb{N}$, $\mathbb{B} = \{0,1\}$. Ein* **n-bit-Blockcode** *ist eine Teilmenge $C \subseteq \mathbb{B}^n$.*

Beispiel für einen **3-bit-Blockcode**: Die Wörter

$$\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}$$

bilden einen Blockcode.
Mit den folgenden Veknüpfungen wird aus $\mathbb{B}^n$ ein Vektorraum:

$\oplus$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

$$\begin{aligned} x \oplus y &= (x_1 \dots x_n) \oplus (y_1 \dots y_n) \\ &= x_1 \oplus y_1 \dots x_n \oplus y_n \\ k \cdot y &= kx_1 \dots kx_n \end{aligned}$$

Noch drei weitere Definitionen sind notwendig:

Definition 1.2 *Der* **Hamming-Abstand** *$d(x,y)$ zweier Codewörter $x, y \in \mathbb{B}^n$ ist die Zahl der verschiedenen Stellen von x und y.*

Zusätzlich zum Hamming-Abstand zwischen zwei Codewörtern definiert man noch den Hammingabstand eines Codes:

Definition 1.3 *Der* **Hamming-Abstand eines Codes** *$C \subseteq \mathbb{B}^n$ ist das Minimum aller Hamming-Abstände zwischen je zwei verschiedenen Codewörtern:*

$$d(C) = min\,\{d(x,y) | x, y \in C, x \neq y\}$$

Definition 1.4 *Das* **Hamming-Gewicht** *$w(x)$ eines Codewortes $x = x_1 \dots x_n \in \mathbb{B}^n$ ist gegeben durch $w(x) = \sum_{i=1\dots n} x_i$*

Beim Hamming-Gewicht werden die Bits eines Codewortes also einfach aufaddiert.

1.3.2 Fehler

Der oben definierte Formalismus kann dazu verwendet werden, Fehler zu erkennen. Definiere dazu noch Fehler:

Definition 1.5 *Sei C ein n-Bit-Code. Ein* **Fehler** *ist ein Codewort $e \in \mathbb{B}^n$ mit $e \neq 0$.*
Ist $w(e) = 1$, so heißt e ein Einfachfehler.
Ist $w(e) = 2$, so heißt e ein Doppelfehler.

Der Code C erkennt einen Fehler e, wenn $\forall x \in C$ gilt: $x \oplus e \notin C$. Das Codewort x wird durch den Fehler e verfälscht zu $x \oplus e$. Dies wird dann als Fehler erkannt, falls $x \oplus e$ kein Codewort aus der Codewortmenge C ist.

Satz 1.1 *Sei C ein n-Bit-Code mit Hammingabstand 2. Dann erkennt C alle Einfachfehler.*

Beweis: e sei Einfachfehler, $x \in C$. Dann ist

$$d(x, x \oplus e) = w(x \oplus x \oplus e) = w(e) = 1$$

Also handelt es sich offenbar um einen Fehler, da der Hammingabstand 2 ist für Codewörter aus C.
Ein Beispiel für einen Code mit Hammingabstand 2 ist der Binärcode mit Paritätsbit $C4,3 \subseteq \mathbb{B}^4$. In der 4. Spalte steht dabei das Paritätsbit. Die Codewörter lauten hier:

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{array}$$

Durch vergleichen zweier Codewörter sieht man, dass immer zwei Bits verschieden sind und damit ein Hammingabstand von 2 vorliegt. Nach Satz (1.1) werden somit alle Einfachfehler erkannt.
Dies ist auch ein Beispiel für einen *linearen Code.*

1.3.3 Lineare Codes und Fehlererkennung

Definition 1.6 *Ein Blockcode $C \subseteq \mathbb{B}^n$ heißt* **linearer Code***, wenn die Summe zweier Codewörter wieder ein Codewort ist, d.h. wenn gilt $\forall x, y \in C : x \oplus y \in C$*

Ein **Beispiel** für einen Blockcode ist der C4,3-Code von oben:

$$\begin{array}{cccccl} & 0 & 0 & 1 & 1 & \\ \oplus & 0 & 1 & 0 & 1 & \\ = & 0 & 1 & 1 & 0 & \in C4,3 \end{array}$$

Definiere nun die Generatormatrix für einen linearen Code. Daraus lassen sich sämtliche Codewörter durch Linearkombination der Zeilen erzeugen.

Definition 1.7 *Sei $C \subseteq \mathbb{B}^n$ ein linearer Code der Dimension k und sei $b_1, \ldots, b_k$ eine Basis von C. Die mit den Basisvektoren als Zeilen gebildete $k \times n$- Matrix*

$$G = \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix}$$

heißt **Generatormatrix** *von C.*

Liegen die Codewörter von C in einem Teilraum des $\mathbb{B}^n$, wobei $\mathbb{B} = \{0, 1\}$, so liegen die möglichen erkennbaren Fehler im Orthogonalraum $C^\perp$. Die **Kontrollmatrix** H sei die Generatormatrix für den Orthogonalraum von C in $\mathbb{B}^n$
Für ein Codewort aus $x \in C$ gilt dann natürlich:

$$x \cdot H^T = 0,$$

da $C \perp C^\perp$ in $\mathbb{B}$ ist.
$x \cdot H^T$ heißt das **Syndrom** von x. Das Syndrom ist $\neq 0$, wenn ein Fehler aufgetreten ist. Dann liegt x im Orthogonalraum $C^\perp$, in dem aber kein Codewort von C liegt.

1.3.4 Konstruktion der Kontrollmatrix

Es soll nun kurz besprochen werden, wie aus der gegebenen Generatormatrix G des Codes $C \subseteq \mathbb{B}^n$ die Kontrollmatrix H konstruiert werden kann, also die Generatormatrix von $C^\perp$.
Für die Kontrollmatrix gilt:

$$G \cdot H^T \equiv 0 \, mod \, 2$$

Die Matrix G läßt sich durch Zeilen- und Spaltenumformung auf die folgende Form bringen:

$$G' = Z \cdot G \cdot S = [E_k | P_{n-k}]$$

wobei E_k die k-dimensionale Einheitsmatrix ist und $Pn - k$ die $(n-k)$-dimensionale Paritätsmatrix heißt, die vom Code abhängig ist.
Die Kontrollmatrix zu G' ist dann gerade:

$$H' = [P_{n-k}^T | E_{n-k}]$$

Durch Rückgängigmachen der Spaltentransformationen erhält man H:

$$H = H' \cdot S^T$$

1.3.5 Hamming-Code

Definition 1.8 *Ein* **n-Bit Hamming-Code** *ist ein linearer Code, mit der Kontrollmatrix H, deren j-ter Spaltenvektor die Binärdarstellung der Zahl j enthält, wobei $j \in \{1, \ldots, n\}$.*

Hamming-Codes haben immer den Hamming-Abstand 3.

Beispiel $n = 5$**:**
Hier lautet die Kontrollmatrix:

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

Mit der in Abschnitt (1.3.4) gezeigten Methode kann man die Generatormatrix G zur Kontrollmatrix H konstruieren. Rechnung zeigt, dass die Generatormatrix durch

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

gegeben ist.

Angenommen, $c \in C$ ist ein Codewort und e ist ein Einfachfehler. Das fehlerhafte Codewort ist dann $x = c \oplus e$. Für das Sydrom von x folgt dann:

$$x \cdot H^T = (c \oplus e) \cdot H^T = e \cdot H^T$$

Das Syndrom gibt dann gerade die Stelle von links nach rechts durchnummeriert an, im Binärcode, an der der Fehler aufgetreten ist.
Dies kann man sich dadurch erklären, dass in H die Spalten von links nach rechts binär durchnummeriert sind. e ist ein Einfachfehler und enthält daher nur an einer einzigen Stelle eine 1 und sonst nur Nullen. $e \cdot H^T$ extrahiert also gerade eine Zeile aus H^T, bzw. eine Spalte aus H. Steht die 1 in e an der k-ten Stelle von links nach rechts gezählt, so extrahiert $e \cdot H^T$ also gerade die k-te Spalte aus H. Da in H gerade die Spalten im Binärcode von links nach rechts durchnummeriert sind, gibt $e \cdot H^T$ genau die Stelle von links nach rechts gezählt an, an der der Fehler im Codewort aufgetreten ist.

Wieder zurück zum **Beispiel**:
Sei $x = (1, 1, 0, 0, 0)$ das fehlerhafte Codewort. Dann ist $x \cdot H^T = (0, 1, 1)$. Damit ist der Fehler also an dritte Stelle von links aufgetreten und damit lautet das ursprüngliche Codewort (1,1,1,0,0), welches ja auch tatsächlich durch G generiert wird.

2 Quantenfehler

Ein cbit wird technisch durch eine makroskopische Anzahl von Atomen realisiert. Quantenmechanische Effekte gehen dabei in den klassischen Grenzfall über, so dass ein cbit nur die diskreten Zustände $|0\rangle$ und $|1\rangle$ annehmen kann. Qubits[2] dagegen werden durch einzelne Atome, Ionen oder Moleküle realisiert. Zum Beispiel kann eine Ionenfalle als Quantencomputer fungieren. Der allgemeine Zustand $|\Psi\rangle$ ist deswegen gegeben durch

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2}$$

mit $\langle\Psi|\Psi\rangle = \alpha^2 + \beta^2 = 1$ und $\alpha, \beta \in \mathbb{R}$.
Dass Quantencomputer aus einzelnen Teilchen aufgebaut werden müssen, macht sie sehr anfällig für äußere Störungen, so dass im Vergleich zu klassischen Computern sehr leicht Fehler auftreten können.
Ohne Fehlerkorrektur wäre es also nicht möglich Quantencomputern eine sinnvolle, praktische Anwendung zuzuweisen.

Im Folgenden werden die benötigten unitären Operatoren zusammengestellt. Nach einem Abschnitt über den Meßprozeß eines Operators wird gezeigt, wie ein fehlerbehafteter Quantenzustand von einem oder mehreren qubits beschrieben werden kann. Mit einigen physikalischen Annahmen, die das Auftreten von Fehlern einschränken, ist die Fehlerkorrektur mit dem in Kapitel 3 erläuterten Fehlerkorrekturcode möglich.

[2] Quantenbits

2.1 Operatoren

Für die Präparation des Ursprungszustandes wird der Hadamardoperator $\mathbf{H}$ notwendig sein. Dieser wirkt auf den 1-qubit-Zustand $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ folgendermaßen:

$$\begin{aligned} \mathbf{H}|\Psi\rangle &= \mathbf{H}(\alpha|0\rangle + \beta|1\rangle) && (3) \\ &= \frac{\alpha}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{\beta}{\sqrt{2}}(|0\rangle - |1\rangle) && (4) \end{aligned}$$

In Matrixdarstellung:

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Der qubit-Fehlerzustand wird durch die Operatoren $\mathbf{X}, \mathbf{Y}$ und $\mathbf{Z}$ dargestellt, mit der Wirkung

$$\begin{aligned} \mathbf{X}|\Psi\rangle &= \mathbf{X}(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle && (5) \\ \mathbf{Z}|\Psi\rangle &= \alpha|0\rangle - \beta|1\rangle && (6) \\ \mathbf{Y}|\Psi\rangle &= \mathbf{ZX}|\Psi\rangle = \alpha|1\rangle - \beta|0\rangle && (7) \end{aligned}$$

In Matrizendarstellung:

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\mathbf{Y} = \mathbf{ZX} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

$\mathbf{X}$ und $\mathbf{Z}$ entsprechen den Paulimatrizen σ_x und σ_z. $\mathbf{Y}$ entspricht der Form nach σ_y und es gilt $-i \cdot \mathbf{Y} = \sigma_\mathbf{y}$.

2.2 Der Prozeß der Messung eines Operators

Betrachte einen n-qubit-Operator $\mathbf{A}$ mit der Eigenschaft $\mathbf{A^2} = \mathbf{1}$ und den Eigenwerten $+1, -1$. Die Projektoren auf die Eigenräume von $\mathbf{A}$ lauten dann:

$$\mathbf{P}_0^\mathbf{A} = \frac{\mathbf{1} + \mathbf{A}}{2} \quad \text{Eigenwert 1} \qquad (8)$$

$$\mathbf{P}_1^\mathbf{A} = \frac{\mathbf{1} - \mathbf{A}}{2} \quad \text{Eigenwert -1} \qquad (9)$$

Zusätzlich zu den n qubits existierte ein weiteres qubit, das als control-bit dient. Kontrollierte Operatoren werden hier geschrieben als

$$\mathbf{cA}$$

Der Operator $\mathbf{A}$ wird also nur auf den Zustand angewendet, wenn das control-bit im Zustand $|1\rangle$ ist.

Jetzt soll der **Prozeß der Messung eines Operators** definiert werden. Dazu sei ein n-qubit-Zustand $|\Psi\rangle$ gegeben. Führe nun die folgende Operation aus:

$$(\mathbf{H} \otimes \mathbf{1})\mathbf{cA}(\mathbf{H} \otimes \mathbf{1})|0\rangle|\Psi\rangle$$

$|0\rangle$ ist das control-bit. $\mathbf{H}$ wirkt jeweils auf das control-bit. $\mathbf{1}$ und $\mathbf{A}$ wirken auf $|\Psi\rangle$, wobei $\mathbf{1}$ die Identität ist und den Zustand nicht verändert. $\mathbf{cA}$ wirkt natürlich nur dann, falls das control-bit im Zustand $|1\rangle$ ist. Beachtet man diese Regeln, dann ergibt sich folgende Rechnung:

$$\begin{aligned} &(\mathbf{H} \otimes \mathbf{1})\mathbf{cA}(\mathbf{H} \otimes \mathbf{1})|0\rangle|\Psi\rangle \\ &= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes \mathbf{1})\mathbf{cA}(|0\rangle + |1\rangle)|\Psi\rangle \\ &= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes \mathbf{1})(|0\rangle + |1\rangle\mathbf{A})|\Psi\rangle \\ &= \frac{1}{2}((|0\rangle + |1\rangle) + (|0\rangle - |1\rangle)\mathbf{A})|\Psi\rangle \\ &= |0\rangle\frac{\mathbf{1} + \mathbf{A}}{2}|\Psi\rangle + |1\rangle\frac{\mathbf{1} - \mathbf{A}}{2}|\Psi\rangle \\ &= |0\rangle\mathbf{P}_0^\mathbf{A}|\Psi\rangle + |1\rangle\mathbf{P}_1^\mathbf{A}|\Psi\rangle \end{aligned}$$

Durch Messung des control-bits wird der Zustand $|\Psi\rangle$ in einen der Eigenräume des Operators $\mathbf{A}$ projiziert. Dieser Prozeß heißt **Messung von A**.

Bei mehreren kommutierenden Observablen $\mathbf{A}, \mathbf{B}, \mathbf{C}$ gilt entsprechend:

$$\begin{aligned} &(\mathbf{H_2H_1H_0} \otimes \mathbf{1})\mathbf{cCcBcA}(\mathbf{H_2H_1H_0} \otimes \mathbf{1})|0\rangle|0\rangle|0\rangle|\Psi\rangle \\ &= \sum_{x_2=0}^{1}\sum_{x_1=0}^{1}\sum_{x_0=0}^{1} |x_2\rangle|x_1\rangle|x_0\rangle\mathbf{P}_{x_2}^\mathbf{C}\mathbf{P}_{x_1}^\mathbf{B}\mathbf{P}_{x_0}^\mathbf{A}|\Psi\rangle \end{aligned}$$

Analog für beliebig viele kommutierende Observablen.

2.3 Physikalische Annahmen und Fehlerdarstellung durch Pauli-Matrizen

Fehler in Quantencomputern entstehen insbesondere dadurch, dass die qubits des Quantencomputers eine Verschränkung mit der Umgebung erfahren.
Die Umgebung sei durch den Zustand $|e\rangle$ gegeben. Betrachte nun ein System, bestehend aus der Umgebung $|e\rangle$ und einem qubit. Die Verschränkung der Umgebung mit dem qubit schreibt man als

$$|e\rangle|0\rangle \rightarrow |e_0\rangle|0\rangle + |e_1\rangle|1\rangle \quad (10)$$
$$|e\rangle|1\rangle \rightarrow |e_2\rangle|0\rangle + |e_3\rangle|1\rangle \quad (11)$$

Aus dem unverschränkten System wird somit ein mit der Umgebung verschränktes System. Gleichungen (10) und (11) entsprechen einer unitären Zeitentwicklung eines Zustands. Diese Art der Störung des Systems heißt **Dekohärenz**.
Mit dem Projektor $\mathbf{P}_x^{\mathbf{Z}} = \frac{\mathbf{1}+(-1)^x\mathbf{Z}}{2}$ können die Gleichungen (10) und (11) zu einer Gleichung zusammengefasst werden:

$$\begin{aligned}|e\rangle|x\rangle \rightarrow &\left([|e_0\rangle\mathbf{1} + |e_1\rangle\mathbf{X}]\,\mathbf{P}_0^{\mathbf{Z}}\right)|x\rangle \\ &+ \left([|e_2\rangle\mathbf{X} + |e_3\rangle\mathbf{1}]\,\mathbf{P}_1^{\mathbf{Z}}\right)|x\rangle\end{aligned}$$

Durch einsetzen von $x = 0$ oder $x = 1$ kann man diese Formel leicht bestätigen.
Die Zeitentwicklung läßt sich durch explizites einsetzen von $\mathbf{P}_{0,1}^{\mathbf{Z}}$ weiter auswerten:

$$\begin{aligned}&\left([|e_0\rangle\mathbf{1} + |e_1\rangle\mathbf{X}]\,\frac{\mathbf{1}+\mathbf{Z}}{2}\right)|x\rangle \\ &+ \left([|e_2\rangle\mathbf{X} + |e_3\rangle\mathbf{1}]\,\frac{\mathbf{1}-\mathbf{Z}}{2}\right)|x\rangle \\ &= (\frac{|e_0\rangle + |e_3\rangle}{2}\mathbf{1} + \frac{|e_1\rangle + |e_2\rangle}{2}\mathbf{X} \\ &+ \frac{|e_2\rangle - |e_1\rangle}{2}\mathbf{Y} + \frac{|e_0\rangle - |e_3\rangle}{2}\mathbf{Z})|x\rangle\end{aligned}$$

Mit

$$|d\rangle = \frac{|e_0\rangle + |e_3\rangle}{2} \qquad |a\rangle = \frac{|e_1\rangle + |e_2\rangle}{2}$$
$$|b\rangle = \frac{|e_2\rangle - |e_1\rangle}{2} \qquad |c\rangle = \frac{|e_0\rangle - |e_3\rangle}{2}$$

folgt schließlich:

$$|e\rangle|x\rangle \rightarrow (|d\rangle\mathbf{1} + |a\rangle\mathbf{X} + |b\rangle\mathbf{Y} + |c\rangle\mathbf{Z})\,|x\rangle$$

Dies gilt natürlich auch für ein qubit, das sich nicht in einem eindeutigen Zustand befindet, sondern im Zustand $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Also

$$|e\rangle|\Psi\rangle \rightarrow (|d\rangle\mathbf{1} + |a\rangle\mathbf{X} + |b\rangle\mathbf{Y} + |c\rangle\mathbf{Z})\,|\Psi\rangle \quad (12)$$

Jede der Matrizen $\mathbf{X}$, $\mathbf{Y}$ und $\mathbf{Z}$ repräsentiert einen bestimmten Fehlertyp:

- $\mathbf{X}$ entspricht einem Bit-Flip-Fehler
- $\mathbf{Z}$ entspricht einem Phasenfehler
- $\mathbf{Y}$ entspricht einem Bit-Flip und Phasenfehler

Dies ist direkt an der Matrizendarstellung erkennbar:

Bit-flip

$$\begin{aligned}\mathbf{X}(\alpha|0\rangle + \beta|1\rangle) &= \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}(\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha|1\rangle + \beta|0\rangle\end{aligned}$$

Phasenfehler

$$\begin{aligned}\mathbf{Z}(\alpha|0\rangle + \beta|1\rangle) &= \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}(\alpha|0\rangle + \beta|1\rangle) \\ &= (\alpha|0\rangle - \beta|1\rangle)\end{aligned}$$

Bit-flip und Phasenfehler

$$\begin{aligned}\mathbf{Y}(\alpha|0\rangle + \beta|1\rangle) &= \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}(\alpha|0\rangle + \beta|1\rangle) \\ &= (-\alpha|1\rangle + \beta|0\rangle)\end{aligned}$$

Erweiterung von (12) auf einen n-qubit-Zustand ergibt

$$|e\rangle|\Psi\rangle_n \rightarrow \sum_{\mu_1=0}^{3} \cdots \sum_{\mu_n=0}^{3} |e_{\mu_1 \ldots \mu_n}\rangle \mathbf{X}^{(\mu_1)} \otimes \cdots \otimes \mathbf{X}^{(\mu_n)} |\Psi\rangle_n. \quad (13)$$

Wobei $\mathbf{X}^{(0)} \equiv \mathbf{1}$, $\mathbf{X}^{(1)} \equiv \mathbf{X}$, $\mathbf{X}^{(2)} \equiv \mathbf{Y}$ und $\mathbf{X}^{(3)} \equiv \mathbf{Z}$.
Jedes qubit kann also durch alle drei möglichen Fehlertypen gleichzeitig verfälscht werden. Der Fehlerzustand (13) besteht also aus einer Superposition von allen möglichen Fehlern, die überhaupt auftreten können.
Dies würde jedoch Fehlerkorrektur, wie sie in Kapitel 3 durchgeführt wird mit den bekannten Codes unmöglich machen. Deshalb geht man davon aus, dass folgende vereinfachende Bedingung erfüllt ist:

> Wenn $|\Psi\rangle$ der Zustand einer kleinen Zahl n wie oben kodierter qubits ist, dann ist die Wahrscheinlichkeit einer Störung dieser qubits so klein, dass die Terme in (13), die sich von $\mathbf{1} \otimes \mathbf{1} \otimes \cdots \otimes \mathbf{1}$ unterscheiden von denen dominiert werden, in denen nur ein einziges $\mathbf{X}^{(\mu_i)}$ sich von $\mathbf{1}$ unterscheidet.

Mit dieser Annahme vereinfacht sich der Fehlerzustand (13) zu

$$|e\rangle|\Psi\rangle_n \rightarrow \left(|d\rangle\mathbf{1} + \sum_{i=1}^{n} (|a_i\rangle\mathbf{X}_i + |b_i\rangle\mathbf{Y}_i + |c_i\rangle\mathbf{Z}_i) \right) |\Psi\rangle_n$$

Es wird also angenommen, dass keine mehrfachen Fehler auftreten. Der Fehlerzustand besteht somit aus einer Superposition von Einzelfehlern. Nach Messung des Systems bricht die Wellenfunktion zusammen, so dass sich das System anschließend in einem Zustand befindet, der keinen Fehler enthält, oder aber sich in einem Zustand befindet, in dem genau ein Bit den falschen Wert angenommen hat.

3 Quantenfehlerkorrektur

3.1 Über die scheinbare Unmöglichkeit der Quantenfehlerkorrektur

Bis 1995 hielt man die Quantenfehlerkorrektur für nicht realisierbar, da man sich kein geeignetes Verfahren vorstellen konnte. Alle klassischen Verfahren, wie auch die hier besprochene *Fehlerkorrektur durch Wiederholung* und der lineare *Hamming-Code* versagen aus dem gleichen Grund.
Um einen Fehler in einem Bit oder Codewort zu korrigieren, muss zunächst der Zustand des Bits oder Codeworts gemessen werden. Bei der Messung eines Quantensystems, also eines Systems aus Qubits, bricht die Wellenfunktion des Systems zusammen. Befindet sich das System in einem Zustand, der aus mehreren superponierten Eigenzuständen besteht, so wird das System bei Messung in genau einen dieser Eigenzustände gebracht. Dies führt dazu, dass die Information, die in der Superposition enthalten war, verloren ist.
Die Vorteile, die das Quantencomputing durch diese Informationen mit sich bringt, wären mit den Methoden der klassischen Fehlerkorrektur nicht nutzbar. Das Benutzen von superponierten Zuständen macht das Quantencomputing aber gerade so effektiv.
Ohne Fehlerkorrektur wären Quantencomputer wiederum auch ohne grossen Nutzen, da die Fehleranfälligkeit hoch ist.

1995 ist dann von P.W. Shor [7] der erste Fehlerkorrekturcode für Quantencomputer vorgestellt worden, der Fehlerkorrektur ohne Zerstörung des Zustandes möglich macht. Erst damit ist eine weiträumige Nutzung von Quantencomputern wieder denkbar geworden.

In Shor's Code wird jedes Qubit durch 9 kodierte qubits ersetzt. Hier soll im Folgenden der später entwickelte, effektivere 5-Qubit-Code vorgestellt werden.

3.2 Das Prinzip der Quantenfehlerkorrektur

Bei dem hier besprochenen Verfahren handelt es sich um einen sogenannten **stabilisierenden Code**. Stellt man sich vor, dass ein System von Qubits als eine Art Speicher fungiert, so können nach einer gewissen Zeitspanne die in Abschnitt (2.3) beschriebenen Fehler auftreten. In regelmässigen Abständen muss dann ein stabilisierender Code ausgeführt werden, der den Fehler korrigiert.
Das Prinzip der Quantenfehlerkorrektur basiert unabhängig vom spezifischen Code immer auf dem gleichen Verfahren. Mehrere paarweise kommutierenden Observablen werden „geschickt“ gewählt. Jedes einzelne qubit, $|0\rangle$ oder $|1\rangle$ wird durch mehrere qubits auf eine bestimmte Art kodiert. Nun wird angenommen, dass der ursprüngliche Zustand verfälscht wurde und zwar so, dass sich das System nun in dem oben besprochenen vereinfachten Fehlerzustand befindet. Durch Anwenden der kommutierenden Observablen wird das System in genau einen Eigenzustand projiziert, der durch die bei der Messung erhaltenen Eigenwerte eindeutig identifiziert ist. Der fehlerhafte Zustand ist nun bekannt und kann korrigiert werden.

3.3 Der 5-qubit-code

Jedes qubit wird durch fünf qubits kodiert. Dies geschieht durch die 4 Operatoren

$$\mathbf{M}_1 = \mathbf{X}_2\mathbf{Z}_3\mathbf{Z}_4\mathbf{X}_5 \quad (13)$$
$$\mathbf{M}_2 = \mathbf{X}_3\mathbf{Z}_4\mathbf{Z}_5\mathbf{X}_1 \quad (14)$$
$$\mathbf{M}_3 = \mathbf{X}_4\mathbf{Z}_5\mathbf{Z}_1\mathbf{X}_2 \quad (15)$$
$$\mathbf{M}_4 = \mathbf{X}_5\mathbf{Z}_1\mathbf{Z}_2\mathbf{X}_3 \quad (16)$$

Für diese Operatoren gilt $\mathbf{M}_i^2 = \mathbf{1}$.

Definiere nun die kodierten Zustände:

$$\begin{aligned} |\bar{0}\rangle &= \frac{1}{4}(\mathbf{1}+\mathbf{M}_1)(\mathbf{1}+\mathbf{M}_2) \\ &\quad (\mathbf{1}+\mathbf{M}_3)(\mathbf{1}+\mathbf{M}_4)|00000\rangle \\ |\bar{1}\rangle &= \frac{1}{4}(\mathbf{1}+\mathbf{M}_1)(\mathbf{1}+\mathbf{M}_2) \\ &\quad (\mathbf{1}+\mathbf{M}_3)(\mathbf{1}+\mathbf{M}_4)|11111\rangle \end{aligned}$$

Jedes $\mathbf{M}_i$ flipt genau 2 qubits. Damit folgt dann: Jeder Term von $|\bar{0}\rangle$ hat eine ungerade Zahl von Nullen und jeder Term von $|\bar{1}\rangle$ hat eine gerade Zahl von Nullen. Damit folgt dann:

$$|\bar{0}\rangle \perp |\bar{1}\rangle$$

Aus $\mathbf{M}_i^2 = \mathbf{M}_i \Rightarrow (\mathbf{1}+\mathbf{M}_i)^2 = 2(\mathbf{1}+\mathbf{M}_i)$. Damit gilt also:

$$\langle\bar{0}|\bar{0}\rangle = 1 \qquad \langle\bar{1}|\bar{1}\rangle = 1$$

Also sind $|\bar{0}\rangle$ und $|\bar{1}\rangle$ orthonormierte Zustände. Die $\mathbf{M_i}$ kommutierten paarweise. Außerdem gilt

$$\begin{aligned} \mathbf{M}_i(\mathbf{1}+\mathbf{M}_i) = (\mathbf{M}_i + \mathbf{M}_i^2) = \mathbf{1}+\mathbf{M}_i \\ \Rightarrow \mathbf{M}_i|\bar{0}\rangle = |\bar{0}\rangle, \qquad \mathbf{M}_i|\bar{1}\rangle = |\bar{1}\rangle \end{aligned}$$

Damit ist der Zustand $|\bar{\Psi}\rangle = \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$ ein Eigenzustand der $\mathbf{M}_i$ mit dem Eigenwert 1.
Ist der Zustand $|\bar{\Psi}\rangle$ fehlerfrei, so liefert die Anwendung der vier Operatoren die vier Eigenwerte 1 und wieder den Zustand $|\bar{\Psi}\rangle$.

Jetzt sei das System in dem fehlerhaften Zustand

$$\left(|d\rangle\mathbf{1} + \sum_{i=1}^{5}\left(|a_i\rangle\mathbf{X}_i + |b_i\rangle\mathbf{Y}_i + |c_i\rangle\mathbf{Z}_i\right)\right)|\bar{\Psi}\rangle.$$

Durch Messung bricht die Wellenfunktion zusammen und das System ist in *einem* der 16 möglichen Zustände. Es soll nun herausgefunden werden in welchem Zustand das System ist. Dann kann der Fehler korrigiert werden.

Dazu ein **Beispiel:** Das System sei im

Zustand $\mathbf{X}_1|\bar{\Psi}\rangle$ und der Zustand sei bekannt. Es gilt $\mathbf{X}_1^2 = \mathbf{1}$. Durch Anwenden von $\mathbf{X}_1$ auf den fehlerhaften Zustand erhält man wegen $\mathbf{X}_1^2|\bar{\Psi}\rangle = |\bar{\Psi}\rangle$ den fehlerfreien Zustand $|\bar{\Psi}\rangle$.

Wende nun direkt nacheinander die Operatoren $\mathbf{M}_i$ auf den Fehlerzustand an. Entweder kommutieren $(-)$ oder antikommutieren $(+)$ die $\mathbf{M}_i$ mit den $\mathbf{X}^{(i)}$. Berechne dazu exemplarische:

$$\begin{aligned}
\mathbf{M}_1\mathbf{X}_1|\bar{\Psi}\rangle &= \mathbf{X}_2\mathbf{Z}_3\mathbf{Z}_4\mathbf{X}_5\mathbf{X}_1|\bar{\Psi}\rangle \\
&= \mathbf{X}_1\mathbf{X}_2\mathbf{Z}_3\mathbf{Z}_4\mathbf{X}_5|\bar{\Psi}\rangle \\
&= \mathbf{X}_1\mathbf{M}_1|\bar{\Psi}\rangle = \mathbf{X}_1|\bar{\Psi}\rangle \\
\mathbf{M}_2\mathbf{X}_1|\bar{\Psi}\rangle &= \mathbf{X}_3\mathbf{Z}_4\mathbf{Z}_5\mathbf{X}_1\mathbf{X}_1|\bar{\Psi}\rangle \\
&= \mathbf{X}_1\mathbf{X}_3\mathbf{Z}_4\mathbf{Z}_5\mathbf{X}_1|\bar{\Psi}\rangle \\
&= \mathbf{X}_1\mathbf{M}_2|\bar{\Psi}\rangle = \mathbf{X}_1|\bar{\Psi}\rangle \\
\mathbf{M}_3\mathbf{X}_1|\bar{\Psi}\rangle &= \mathbf{X}_4\mathbf{Z}_5\mathbf{Z}_1\mathbf{X}_2\mathbf{X}_1|\bar{\Psi}\rangle \\
&= -\mathbf{X}_1\mathbf{X}_4\mathbf{Z}_5\mathbf{Z}_1\mathbf{X}_2|\bar{\Psi}\rangle \\
&= -\mathbf{X}_1\mathbf{M}_3|\bar{\Psi}\rangle = -\mathbf{X}_1|\bar{\Psi}\rangle \\
\mathbf{M}_4\mathbf{X}_1|\bar{\Psi}\rangle &= \mathbf{X}_5\mathbf{Z}_1\mathbf{Z}_2\mathbf{X}_3\mathbf{X}_1|\bar{\Psi}\rangle \\
&= -\mathbf{X}_1\mathbf{X}_5\mathbf{Z}_1\mathbf{Z}_2\mathbf{X}_3|\bar{\Psi}\rangle \\
&= -\mathbf{X}_1\mathbf{M}_4|\bar{\Psi}\rangle = -\mathbf{X}_1|\bar{\Psi}\rangle
\end{aligned}$$

Dabei wurden die folgenden Relationen benutzt:

$$[\mathbf{X},\mathbf{Y}]_+ = [\mathbf{X},\mathbf{Z}]_+ = [\mathbf{X},\mathbf{Y}]_+ = 0$$
$$[\mathbf{X},\mathbf{X}]_- = [\mathbf{Y},\mathbf{Y}]_- = [\mathbf{Z},\mathbf{Z}]_- = 0$$

Führt man sämtliche Berechnungen aus, so ergibt sich die folgende Tabelle:

	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_3$	$\mathbf{M}_4$
$\mathbf{X}_1$	$-$	$-$	$+$	$+$
$\mathbf{Y}_1$	$-$	$+$	$+$	$+$
$\mathbf{Z}_1$	$-$	$+$	$-$	$-$
$\mathbf{X}_2$	$-$	$-$	$-$	$+$
$\mathbf{Y}_2$	$+$	$-$	$+$	$+$
$\mathbf{Z}_2$	$+$	$-$	$+$	$-$
$\mathbf{X}_3$	$+$	$-$	$-$	$-$
$\mathbf{Y}_3$	$+$	$+$	$-$	$+$
$\mathbf{Z}_3$	$-$	$+$	$-$	$+$
$\mathbf{X}_4$	$+$	$+$	$-$	$-$
$\mathbf{Y}_4$	$+$	$+$	$+$	$-$
$\mathbf{Z}_4$	$-$	$-$	$+$	$-$
$\mathbf{X}_5$	$-$	$+$	$+$	$-$
$\mathbf{Y}_5$	$+$	$+$	$+$	$+$
$\mathbf{Z}_5$	$+$	$-$	$-$	$+$

Die Eigenwerte bei der Messung der vier Operatoren sind gegeben durch:

	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_3$	$\mathbf{M}_4$
$\mathbf{X}_1$	1	1	-1	-1
$\mathbf{Y}_1$	1	-1	-1	-1
$\mathbf{Z}_1$	1	-1	-1	1
$\mathbf{X}_2$	1	1	1	-1
$\mathbf{Y}_2$	-1	1	-1	-1
$\mathbf{Z}_2$	-1	1	-1	1
$\mathbf{X}_3$	-1	1	1	1
$\mathbf{Y}_3$	-1	-1	1	-1
$\mathbf{Z}_3$	1	-1	1	-1
$\mathbf{X}_4$	-1	-1	1	1
$\mathbf{Y}_4$	-1	-1	-1	1
$\mathbf{Z}_4$	1	1	-1	1
$\mathbf{X}_5$	1	-1	-1	1
$\mathbf{Y}_5$	-1	-1	-1	-1
$\mathbf{Z}_5$	-1	1	1	-1

Bei Messung der $\mathbf{M}_i$ wird also jeder der 16 möglichen Fehlerzustände eindeutig identifiziert. Durch Anwenden des entsprechenden Fehleroperators ergibt sich der fehlerfreie Zustand, wie in folgendem Beispiel demonstriert.

Beispiel: Angenommen der Fehlerzustand ist durch die Messung der vier Operatoren $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$ und $\mathbf{M}_4$ mit den Eigenwerten 1,1,-1 und -1 als $|F\rangle = |a_1\rangle\mathbf{X}_1|\bar{\Psi}\rangle$ identifiziert worden. Wobei $|a_1\rangle$ aus der Verschränkung mit der Umgebung resultiert. Dann folgt: $\mathbf{X}_1|F\rangle = |a_1\rangle\mathbf{X}_1^2|\bar{\Psi}\rangle = |a_1\rangle|\bar{\Psi}\rangle$ und damit ist der fehlerfreie Zustand wiederhergestellt.

Literatur

[1] N.D. Mermin, Lecture Notes on Quantum Computation, Cornell University Press, 2005

[2] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2002

[3] Lang, Skript Algorithmen, FH Flensburg

[4] E. Knill et al., Introduction to Quantum Error Correction, arXiv:quant-ph/0207170 v1, 2002

[5] A. Steane, Quantum Errors Corrected, Nature 432 (2004)

[6] J. Chiaverini et al., Realization of quantum error correction, Nature 432 (2004)

[7] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, Phys. Rev. A 52,4 (1995)