Fehlerkorrektur

Anzahl korrigierbarer Fehler: $k = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ — $\lfloor\ \rfloor$ = abgerundet / $k \leq (d_{min}-1):2$

↳ Korrigiert zum Code mit der höchsten Wahrscheinlichkeit

In der Mitte liegender Code, kann nicht korrigiert werden, aber als fehlerhaft erkannt werden

Anzahl Prüfbits (p) um einen 1-Bit Fehler in K Datenbits zu korrigieren: $I(p) = \log_2(N+1) \approx \log_2(K+1)$, $p \geq \log_2(K+p+1)$ — aufrunden

z.B. $K = 200$
- Näherung: $\log_2(200+1) = 7.6\text{ bit} \approx 8\text{ bit}$
- Versuchen mit $p = 8$: $8 \geq \log_2(200+8+1) = 7.7\text{bit} \approx 8\text{ bit}$ ✓
- Coderate R: $200/208 = 0.96$

Bildung der Generatormatrix

Generatormatrix (G) setzt sich aus Paritätsmatrix (P) und Einheitsmatrix (I) zusammen.

Die Paritätsbits müssen voneinander unabhängig sein und der Code muss linear sein.

Bei $d_{min} = 3$ muss jede Zeile mind. 3 Einsen aufweisen. (Einheitsmatrix 1-Mal Eins, Paritätsmatrix mind 2-Mal Eins)

z.B. c = {[000000], [011100], [101010], [110001], [110110], [101101], [011011], [000111]} ⓘ Immer grösst mögliche Einheitsmatrix bilden

↳ In diesem Beispiel 3x3 (100, 010, 001)

$$\underbrace{\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}}_{\text{N-K Spalten}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{K Spalten}} \Big\} \text{ K Zeilen}$$

Paritätsmatrix — Einheitsmatrix

G = Generatormatrix N = 6 und K = 3

Einsen immer diagonal von oben links nach unten rechts. Einheitsmatrix immer quadratisch.

Bildung der Prüfmatrix

Paritätsmatrix links in Generatormatrix → Paritätsmatrix unten in Prüfmatrix (und umgekehrt)

Generatormatrix (G) → Prüfmatrix (H^T)

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad H^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

ⓘ (7,4), N = 7, K = 4

Generatormatrix (G) → Prüfmatrix (H^T)

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow H^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Encoder

Durch multiplikation des Datenvektors $\underline{u}$ mit der Generatormatrix G wird das Codewort $\underline{c}$ erzeugt.

z.B. $\underline{u}$ = 0101

Codewort kann so lange wie Einheitsmatrixbreite sein.

Generatormatrix (G)

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Es müssen nur Linien beachtet werden bei welchen im Codewort $\underline{u}$ eine 1 vorkommt.

XOR: Gerade Anzahl 1 XOR = 0; Ungerade Anzahl 1 XOR = 1

$= \underbrace{1\ 1\ 0}_{\text{Prüfbits}}\ \underbrace{0\ 1\ 0\ 1}_{\text{Codewort } \underline{u}\text{ / Infobits}}$ → Codewort $\underline{c}$ welches gesendet wird (beim senden können Fehler auftreten)

$+ \ 0\ 0\ 0\ 0\ 0\ 1\ 0$ → Fehler werden mit dem Fehlervektor $\underline{e}$ beschrieben

$= 1\ 1\ 0\ 0\ 1\ 1\ 1$ → Empfangenes Bitmuster $\underline{\tilde{c}}$ ($\underline{\tilde{c}} = \underline{c} + \underline{e}$)

Decoder

Durch multiplikation des empfangenen Bitmusters $\underline{\tilde{c}}$ mit der Prüfmatrix H^T wird das Syndrom bestimmt.

↳ $\underline{s} = 000$: Kein Fehler, $\underline{s} \neq 000$: Der Index von $\underline{s}$ in der Prüfmatrix H^T ist die Position des zu korrigierenden Fehlers.

	1	1	0	0	1	0	1	⟶ Gesendetes Codewort $\underline{c}$
+	0	0	0	0	0	1	0	⟶ Fehlervektor $\underline{e}$
=	1	1	0	0	1	1	1	⟶ Empfangenes Bitmuster $\underline{\tilde{c}}$
+	0	0	0	0	0	1	0	⟶ Korrektur = Fehlervektor
						└		⟶ Index ($\underline{s}$) = 6
=				0	1	0	1	⟶ Daten decodiert $\underline{\hat{u}}$

Prüfmatrix (H^T)

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} * \begin{array}{|ccc|l} 1 & 0 & 0 & [1] \\ 0 & 1 & 0 & [2] \\ 0 & 0 & 1 & [3] \\ 1 & 1 & 0 & [4] \\ 0 & 1 & 1 & [5] \\ 1 & 1 & 1 & [6] \longrightarrow \text{Index } (\underline{s}) \text{ in } H^T \\ 1 & 0 & 1 & [7] \end{array}$$

$1 \quad 1 \quad 1$ ⟶ Syndrom $\underline{s}$

(!) Wie viele verschiedene Syndrome gibt es im Code?

$N = 7,\ K = 4,\ N-K = 7-4 = 3 \longrightarrow$ max. $2^3 = 8$ verschiedene Syndrome

(!) Wie viele Syndrome sind notwendig um alle Fehler zu korrigieren?

$$\text{Anzahl Syndrome} = \sum_{i=0}^{\left\lfloor \frac{d_{min}-1}{2} \right\rfloor} \binom{N}{i}$$