

Fehlerkorrektur

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

\hookrightarrow 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: $l(p) = \log_2(N+1) \approx \log_2(K+1)$, $p \geq \log_2(K+p+1)$ aufgerunden

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

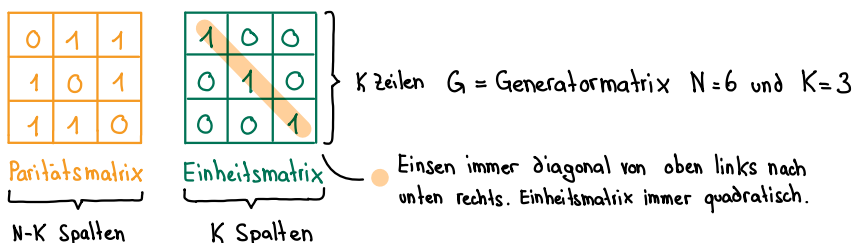
Generatormatrix (G) setzt sich aus Paritätsmatrix (P) und Einheitsmatrix (I) zusammen.

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

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

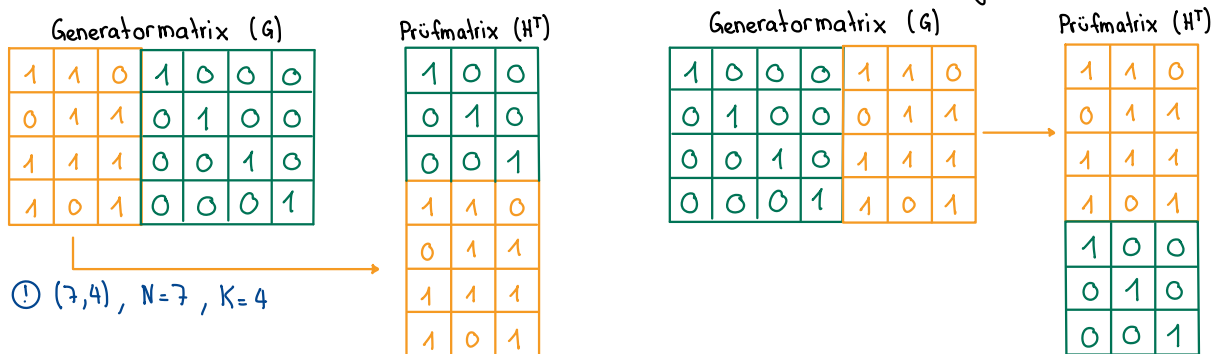
z.B. $C = \{[000000], [011100], [101010], [110001], [110110], [101101], [011011], [000111]\}$ ⚠ Immer grösst mögliche Einheitsmatrix bilden

\hookrightarrow In diesem Beispiel 3×3 (100, 010, 001)



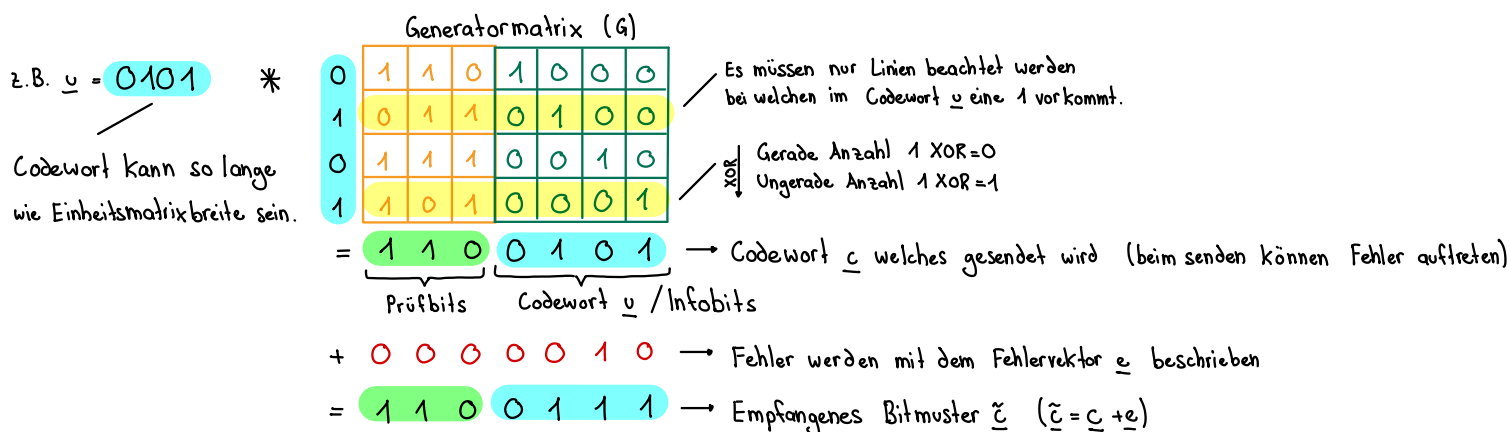
Bildung der Prüfmatrix

Paritätsmatrix links in Generatormatrix \rightarrow Paritätsmatrix unten in Prüfmatrix (und umgekehrt)



Encoder

Durch Multiplikation des Datenvektors u mit der Generatormatrix G wird das Codewort c erzeugt.



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.

$\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 1 & 0 & 1 \rightarrow \text{Gesendetes Codewort } \underline{c} \\ + & 0 & 0 & 0 & 0 & 1 & 0 \rightarrow \text{Fehlervektor } \underline{e} \\ = & 1 & 1 & 0 & 0 & 1 & 1 \rightarrow \text{Empfangenes Bitmuster } \underline{\tilde{c}} \quad * \\ + & 0 & 0 & 0 & 0 & 0 & 1 \rightarrow \text{Korrektor = Fehlervektor} \\ & & & & \text{Index } (\underline{s}) = 6 \\ = & 0 & 1 & 0 & 1 & & \rightarrow \text{Daten decodiert } \underline{\hat{u}} \end{array}$

Prüfmatrix (H^T)

1	1	0	0	[1]
1	0	1	0	[2]
0	0	0	1	[3]
0	1	1	0	[4]
1	0	1	1	[5]
1	1	1	1	[6] → Index (s) in H^T
1	1	0	1	[7]

1 1 1 → Syndrom s

- ❗ Wie viele verschiedene Syndrome gibt es im Code?
 $N = 7, K = 4, N - K = 7 - 4 = 3 \rightarrow \text{max. } 2^3 = 8 \text{ verschiedene Syndrome}$
- ❗ Wie viele Syndrome sind notwendig um alle Fehler zu korrigieren?

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