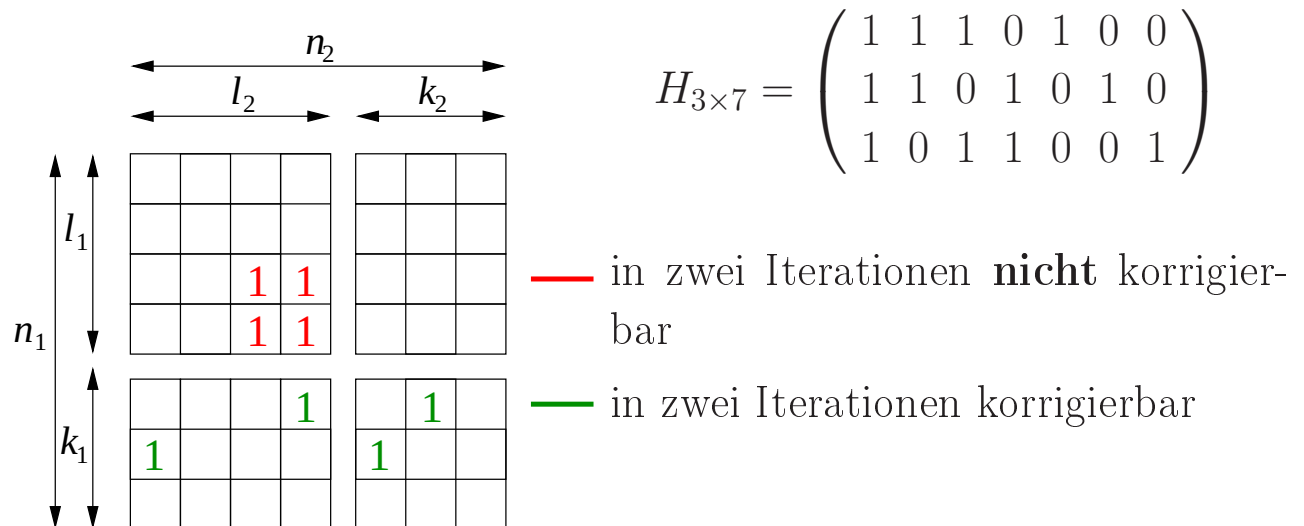


Serielle Verkettung von (7,4)HKs $\rightarrow$ (49,16,9)Produkt-kode:



Möglichkeiten der (**hard-decision**) Dekodierung:

1. Syndromdekodierung mit $s = H \cdot b^T$ über Zeilen/Spalten
2. Syndromdekodierung wie 1. über Zeilen/Spalten/Zeilen/Spalten
 $\rightarrow$ Iterative Dekodierung mit hard-decision!
3. ML Dekodierung

Ergebnisse:

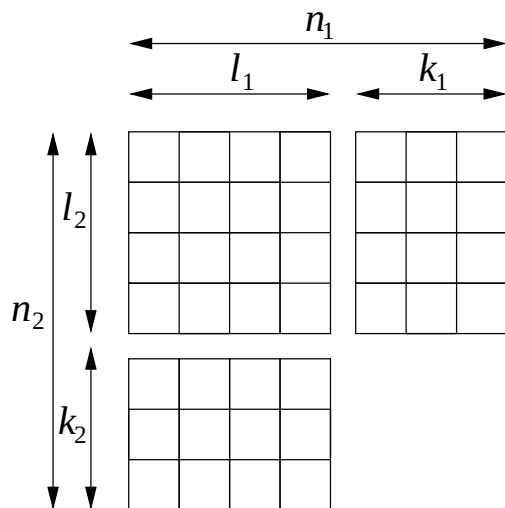
- $f_k = 4$ nur mit 3. erfolgreich (1. mit 95%, 2. mit 99,3%)
- 2. viel schneller als 3. und wesentlich besser als 1.
- Vorteil von Kodeverkettung: auch für $w(e) > f_k$ korrekte Rekonstruktion möglich, Ergebnisse für $p_s = 0.08$ (SBK):

$w(e)$	1.	2.	3.
5	82%	96%	100%
6	58%	86%	97%
7	52%	70%	92%

$\rightarrow$ Iterative Dekodierung mit soft-decision?

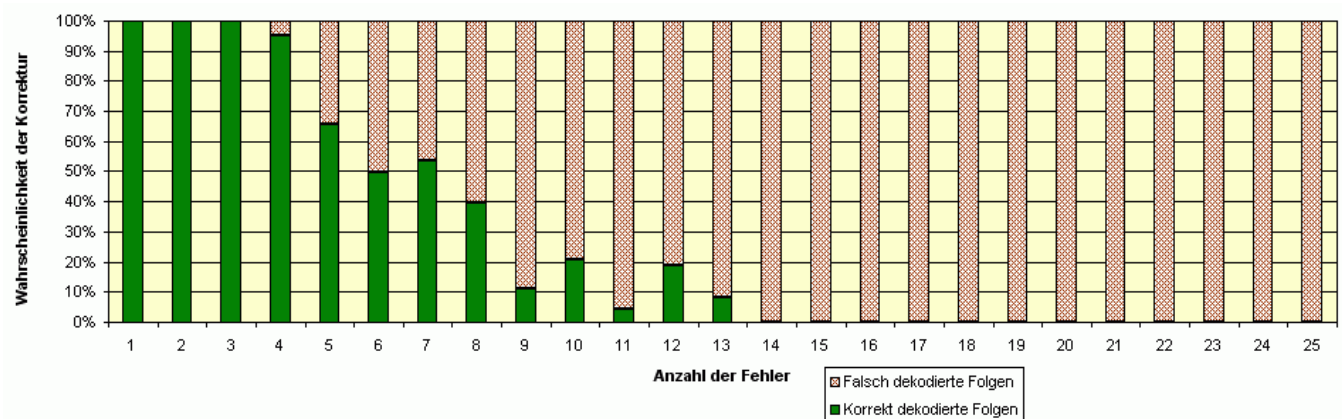
(4.a,b) Parallele Verkettung von (7,4)HKs, $d_{min} \geq 5$:

Abbruch abhängig von L_e Konvergenz!

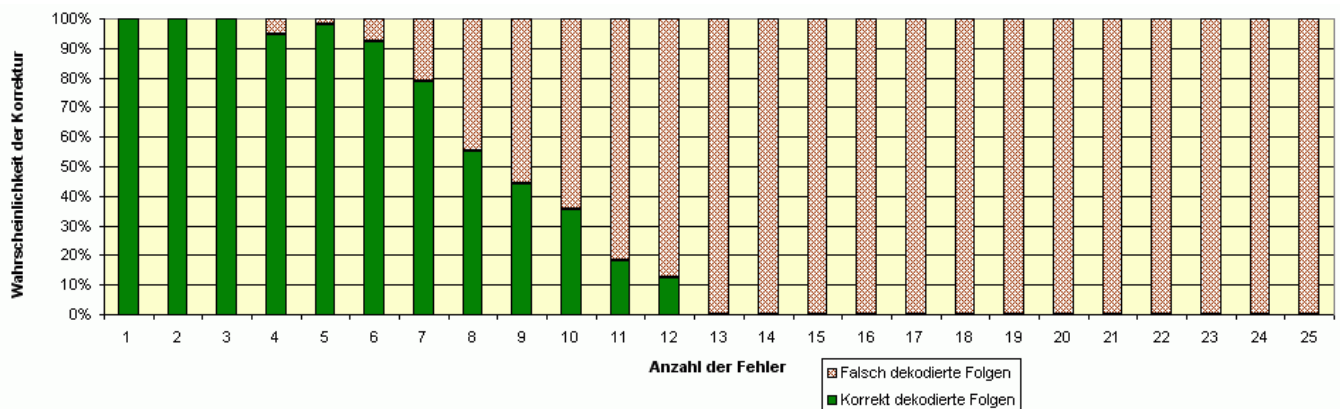


$$H_{3 \times 7} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Willkürliche Wahl einer Bestimmungsgleichung :



Durchschnitt der Ergebnisse aller Bestimmungsgleichungen :



(5.) Umsetzung der parallelen Verkettung in eine Kontrollmatrix $H_{24 \times 40}$:

[illegible]

- $d_{min} = 5$, z. B. Addition der Spalten 25, 27, 28, 36 und 40 führt auf das Null-Syndrom
- 4er Zyklen in H ! Viele Spalten mit Gewicht 1!
- Abbruch abhängig von $s = H \cdot b_{korr}^T = \mathbf{0}$
- min-sum Dekodierung

Ergebnisse:

$w(e)$	3.	4.a	4.b	5.
4	100%	96%	95%	99%
5	100%	67%	98%	98%
6	97%	50%	92%	96%
7	92%	53%	79%	93%

Zu den Ergebnissen von 5.:

- durchschnittlich 1.6 Iterationen
- max. 30 (-200) Iterationen
- Anteil an Falschkorrekturen ist größer als Anteil an Dekodierungsversagen! Kippt mit nur wenigen (z. B. 5) Iterationen!

Zum Vergleich (7,4)HK mit $H_{3 \times 7} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$

$w(e)$	5.
1	97%
2	70%
3	30%

durchschnittlich 1.3 Iterationen

Ergebnisse

(7,4)HK
(40,16)HK(LDPC)
5 Iterationen
30 Iterationen

