

Beispiel zur Korrektur von 8-Bit-Wörtern

Es bestehe ein 8-Bit-Code (wie ANSI) mit folgenden Eigenschaften

- der Code ist *dicht*: es gibt mehr als 128 Zustände
- der Code hat deshalb einen *Hamming-Abstand* (d) von 1
- der Code ist minimal so zu erweitern, damit 1-Bit-Fehler korrigiert werden können.

Die geraden Teil-Paritäten könnten wie folgt berechnet werden (Fantasie):

B7	B6	B5	B4	B3	B2	B1	B0	p0	p1	p2	p3	p4	p5
1	0	1	0	0	1	1	0	0	0	0	0	1	1
				p0									
p1													
		p2						p2					
		p3						p3					
p4						p4							
p5				p5									
Andere Nutz-Wörter mit Hamming-Abstand 1 > Hamming-Abstand 3													
1	0	1	0	0	1	1	1	1	0	1	0	1	1
1	0	1	0	0	1	0	1	0	0	1	1	1	1

Jede 1-Bit-Änderung des Nutz-Wortes (B0 . . . B7) bewirkt

- eine Änderung von mindestens 2 Paritätsbits (p0 . . . p5)
- einen d des 14-Bit-Code-Wortes um mindestens 3. Mit einem $d = 3$ lassen sich korrigieren $\text{int}((d-1)/2) = 1$ Bit.

Der grosse Aufwand an redundanten Bits ist ein Grund, weshalb in spez. Speicherbausteinen und Prozessoren tatsächlich nur 1-Bit-Fehler korrigiert werden. 2-Bit-Fehler werden erkannt. Das Risiko ihres Auftretens ist in den meisten Bausteinen wirtschaftlich tragbarer als die Mehrkosten von ECC auf 2-Bit-Ebene.