Zur Mehrbitfehlerkorrektur mit zyklischen Codes (z. B. BCH-Codes oder Reed-Solomon-Codes)
(siehe auch **„Grundkurs Codierung“**, Unterkapitel 2.2 und 3.7)

Hierfür sind endliche Körper, die **Galoiskörper** oder **Galoisfelder**, hilfreich. Ein algebraischer Körper besteht aus einer endlichen oder unendlichen Menge von Elementen, bei denen die Addition, Subtraktion, Multiplikation und Division wieder ein Element der Menge ergibt. Außerdem enthält sie ein Nullelement (das kann z. B. die „0“ sein) und ein Einselement (das kann z. B die „1“ sein). Bei den „Alltagszahlen“ sind es etwa die rationalen Zahlen oder die reellen Zahlen, die diese Eigenschaften aufweisen.

Galoisfelder lassen sich aus Nullstellen irreduzibler Polynome über Primzahlenkörpern Z_p , mit p = 2, 3, 5, . (Primzahle) aufbauen. Ein irreduzibles Polynom über Z_p lässt sich nicht als Produkt von Polynomen geringerer Ordnung über Z_p darstellen (es ist in dieser Eigenschaften den Primzahlen ähnlich).

Irreduzible Polynome über Z_p gibt es mit jedem Grad m = 1, 2, 3, 4, 5, ...

Für die Zwecke der Fehler-korrigierenden Codes ist der **Primzahlenkörper** Z_2 besonders gut geeignet (warum?).

Ein irreduzibles Polynom mit dem Grad m = 3 über Z_2 ist $f(x) = x^3 + x + 1$ (das einzige weitere mit m = 3 ist $f(x) = x^3 + x^2 + 1$). Es liefert mit $f(\alpha) = \alpha^3 + \alpha + 1 = 0$ die Nullstelle α. Wegen $\alpha^3 = + \alpha + 1$ ist daher die Potenz α^3 mit $\alpha + 1$ verknüpft. Daraus ergibt sich für die Potenzen von α die folgende Tabelle:

Galoisfeld GF(2^3)

Nr.	Element		alternativ		als Polynom in α		Kurzform
1	0	=	0	=	0	=	000
2	α^0	=	1	=	$0\cdot\alpha^2+0\cdot\alpha^1+1\cdot\alpha^0$	=	001
3	α^1	=	$\alpha^0\cdot\alpha$	=	$0\cdot\alpha^2+1\cdot\alpha^1+0\cdot\alpha^0$	=	010
4	α^2	=	$\alpha^1\cdot\alpha$	=	$1\cdot\alpha^2+0\cdot\alpha^1+0\cdot\alpha^0$	=	100
5	α^3	=	$\alpha^2\cdot\alpha$	=	$0\cdot\alpha^2+1\cdot\alpha^1+1\cdot\alpha^0$	=	011
6	α^4	=	$\alpha^3\cdot\alpha$	=	$1\cdot\alpha^2+1\cdot\alpha^1+0\cdot\alpha^0$	=	110
7	α^5	=	$\alpha^4\cdot\alpha$	=	$1\cdot\alpha^2+1\cdot\alpha^1+1\cdot\alpha^0$	=	111
8	α^6	=	$\alpha^5\cdot\alpha$	=	$1\cdot\alpha^2+0\cdot\alpha^1+1\cdot\alpha^0$	=	101
9=2	α^7	=	$\alpha^6\cdot\alpha$	=	$0\cdot\alpha^2+0\cdot\alpha^1+1\cdot\alpha^0$	=	001
10=3	α^8	=	$\alpha^7\cdot\alpha$	=	$0\cdot\alpha^2+1\cdot\alpha^1+0\cdot\alpha^0$	=	010
....	....	...	....	....	....	....	....

Nimmt man die „0“ hinzu, enthält sie mit den ersten 7 Potenzen von α einen endlichen Körper, das Galoisfeld GF(2^3).

Die **Stellenzahl** der damit aufbaubaren Codes ist immer $n= 2^m – 1$, wobei m den Grad des höchsten im Generatorpolynom g(x) vorkommenden irreduziblen Polynoms über Z_2 angibt.

Hier einige Polynome über Z_2 für die Grade m = 1, 2 , 3, 4:

$m = 1: \; x^1+1$
$m = 2: \; x^2+x^1+1$
$m = 3: \; x^3+x^1+1 \;$ und $\; x^3+x^2+1$
$m = 4: \; x^4+x^1+1 \;$ und $\; x^4+x^3+x^2+x^1+1 \;$ und $\; x^4+x^3+1$

Ein Galoisfeld GF(2^4) kann über das definierende Polynom $f(x) = x^4 + x + 1$ für das Element β wie folgt aufgebaut werden:

Nr.	Element		alternativ		als Polynom in β		Kurzform
1	0	=	0	=	0	=	0000
2	β^0	=	1	=	$0\cdot\beta^3+0\cdot\beta^2+0\cdot\beta^1+1\cdot\beta^0$	=	0001
3	β^1	=	$\beta^0\cdot\beta$	=	$0\cdot\beta^3+0\cdot\beta^2+1\cdot\beta^1+0\cdot\beta^0$	=	0010
4	β^2	=	$\beta^1\cdot\beta$	=	$0\cdot\beta^3+1\cdot\beta^2+0\cdot\beta^1+0\cdot\beta^0$	=	0100
5	β^3	=	$\beta^2\cdot\beta$	=	$1\cdot\beta^3+0\cdot\beta^2+0\cdot\beta^1+0\cdot\beta^0$	=	1000
6	β^4	=	$\beta^3\cdot\beta$	=	$0\cdot\beta^3+0\cdot\beta^2+1\cdot\beta^1+1\cdot\beta^0$	=	0011
7	β^5	=	$\beta^4\cdot\beta$	=	$0\cdot\beta^3+1\cdot\beta^2+1\cdot\beta^1+0\cdot\beta^0$	=	0110
8	β^6	=	$\beta^5\cdot\beta$	=	$1\cdot\beta^3+1\cdot\beta^2+0\cdot\beta^1+0\cdot\beta^0$	=	1100
9	β^7	=	$\beta^6\cdot\beta$	=	$1\cdot\beta^3+0\cdot\beta^2+1\cdot\beta^1+1\cdot\beta^0$	=	1011
10	β^8	=	$\beta^7\cdot\beta$	=	$0\cdot\beta^3+1\cdot\beta^2+0\cdot\beta^1+1\cdot\beta^0$	=	0101
11	β^9		$\beta^8\cdot\beta$		$1\cdot\beta^3+0\cdot\beta^2+1\cdot\beta^1+0\cdot\beta^0$		1010
12	β^{10}		$\beta^9\cdot\beta$		$0\cdot\beta^3+1\cdot\beta^2+1\cdot\beta^1+1\cdot\beta^0$		0111
13	β^{11}		$\beta^{10}\cdot\beta$		$1\cdot\beta^3+1\cdot\beta^2+1\cdot\beta^1+0\cdot\beta^0$		1110
14	β^{12}		$\beta^{11}\cdot\beta$		$1\cdot\beta^3+1\cdot\beta^2+1\cdot\beta^1+1\cdot\beta^0$		1111
15	β^{13}		$\beta^{12}\cdot\beta$		$1\cdot\beta^3+1\cdot\beta^2+0\cdot\beta^1+1\cdot\beta^0$		1101
16	β^{14}		$\beta^{13}\cdot\beta$		$1\cdot\beta^3+0\cdot\beta^2+0\cdot\beta^1+1\cdot\beta^0$		1001
17	β^{15}		$\beta^{14}\cdot\beta$		$0\cdot\beta^3+0\cdot\beta^2+0\cdot\beta^1+1\cdot\beta^0$		0001

Der damit konstruierbare Code hat die Länge $n = 2^4 - 1 = 15$. Er ist zur Korrektur von t_{ko} = 1, 2, oder 3 Fehlern (Voraussetzung immer: HD-Demodulation. Für Soft-Decision SD kann man bessere = geringere Restfehlerraten erwarten).

Aber Vorsicht: Für Codes zur Korrektur von t_{kor} Fehlern sind nur ganz bestimmte Kombinationen von irreduziblen Polynomen geeignet (Begründung siehe „Grundkurs Codierung“, Unterkapitel 3.7).

Hier der prinzipielle Weg zur Erzeugung von zyklischen Codewörtern und zur Decodierung der HD *)-Empfangswörter w(x):

Nullstellen im GF(2^m)	$a_1, a_2, ... , a_t$
irreduzible Teilpolynome hierzu	$g_1(a_1) = 0,\ g_2(a) = .0, ..., g(_t) = 0$
Generatorpolynom	$g(x) = g_1(x)\cdot g_2(x)\cdot ..... g_t(x)$
Infopolynom	u(x)
Codewortpolynom	$u(x)\cdot x^{\text{grad } g(x)} + [u(x)\cdot x^{\text{grad } g(x)}]$ MOD g(x)
Fehlerpolynom	e(x)
Empfangswortpolynom	w(x) = v(x) + e(x)
Syndrompolynom	s(x) = w(x) MOD g(x) = e(x) MOD g(x)
Werte von s(x) an den Nullstellen:	
- für x= a_1	$s(a_1) = v(a_1) + e(a_1) = 0 + e(a_1) = e(a_1)$
- für x = a_2	$s(a_2) = v(a_2) + e(a_2) = 0 + e(a_2) = e(a_2)$
-.....	.....
- für x = a_t	$s(a_t) = v(a_t) + e(a_t) = 0 + e(a_t) = e(a_t)$

*) HD = Abkürzung für **Hard-Decision (-Entscheidung)**. Hierbei wird das empfangene Signal-Bit, welches gegenüber dem gesendeten Codewort-Bit durch das Rauschsignal verfälscht ist, durch die "harten" Entscheidungen

- positives Empfangssignal → "0"
- negatives Empfangssignal → "1"

in einfach weiter verarbeitbare Bitfolgen abgebildet.

Allerdings verliert man bei diesem groben Vorgehen einige statistische Informationen, welche die Fehlerkorrektur-Leistung eines Codes begrenzen. Die **SD-Entscheidung** (=Soft-Decision-Entscheidung) nutzt diese zusätzlichen statistischen Informationen zu einer bedeutenden Leistungsverbesserung, ist allerdings mit wesentlich größerem Rechenaufwand verbunden.