Fehlererkennung

Jedes Codewort besteht aus N Bits, wovon K Informationsbits sind und P Prüfbits. $N = K + P$

Mit N Bits gibt es: 2^N mögliche Bitmuster, 2^K gültige Codewörter, Restliche ungültige Bitmuster für Fehlererkennung

Fehlererkennung mit Parity

Even Parity: Anzahl 1er inkl. Parity-Bit ist gerade (Nur Even-Parity ermöglicht lineare Codes.)

Parity	Daten							
1	0	1	0	1	1	0	1	1

Odd Parity: Anzahl 1er inkl. Parity-Bit ist ungerade

Parity	Daten							
0	0	1	0	1	1	0	1	1

1-Bit Arithmetik

Addition / Subtraktion

a	b	r
0	0	0
0	1	1
1	0	1
1	1	0

$r = a \oplus b$ (XOR)

Multiplikation

a	b	r
0	0	0
0	1	0
1	0	0
1	1	1

$r = a \cdot b$ (AND)

1-Bit Polynom-Arithmetik

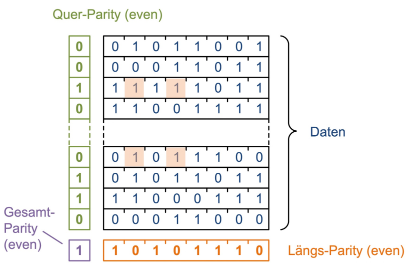
Bei CRC werden die Bits als Koeffizienten eines Polynoms aufgefasst.

Binäres Datenwort $\underline{u} = (101001)$ wird zum Polynom $U(z) = 1 \cdot z^5 + 0 \cdot z^4 + 1 \cdot z^3 + 0 \cdot z^2 + 0 \cdot z^1 + 1 \cdot z^0 = z^5 + z^3 + 1$

Addition/Subtraktion (1-Bit Arithmetik): $(z^3 + z^2 + 1) \pm (z^2 + z + 1) = z^3 \cdot (1 \pm 0) + z^2 \cdot (1 \pm 1) + z^1 \cdot (0 \pm 1) + z^0 \cdot (1 \pm 1) = z^3 + z$

Multiplikation: $(z^2 + z + 1) \cdot (z^2 + z) = (z^4 + z^3) + (z^3 + z^2) + (z^2 + z) = z^4 + (z^3 + z^3) + (z^2 + z^2) + z = z^4 + z^3 \cdot (1+1) + z^2 \cdot (1+1) + z = z^4 + z$

2D Parity



Nicht erkennbare 4-Bit Fehler

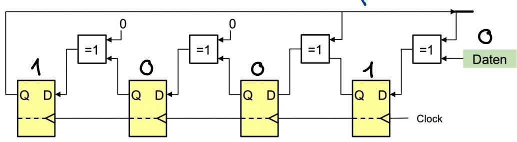
$\underline{e} = d_{min} - 1 = 3$ (maximal 3 Fehler erkennbar)

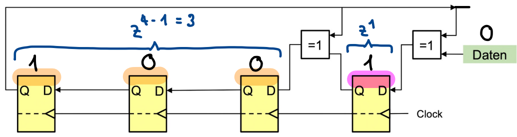
$d_{min} = 4$ $(e+1)$

CRC Hardware Implementierung

z.B. $1 \cdot z^4 + 0 \cdot z^3 + 0 \cdot z^2 + 1 \cdot z^1 + 1 \cdot z^0 = \underbrace{z^4 + z + 1}_{} = 10011_b$

2 XOR Gatter (Anzahl z = Anzahl XOR Gatter)





Cyclic Redundancy Check (CRC)

Ein Bitfehler soll sich auf möglichst viele Bits der Prüfsumme (CRC) auswirken. Fehler wird nicht erkannt wenn der Fehlervektor genau dem Generator-Polynom oder einem Vielfachem entspricht. (z.B. Codewort 16 Bit davon 12 Bit Nutzdaten, 16-12 = 4 → $z^4 + z + 1$)

z.B. Generator-Polynom: $\underline{g} = z^4 + z + 1 = 10011$ $m = 4$, Nutzdatenwort $\underline{u} = 1000010$

Encoder:

1. m Nullen anhängen $(z^m = z^4)$ = 10000100000
2. Polynomdivision

```
  1 0 0 0 0 1 0 0 0 0 0 : 1 0 0 1 1
- 1 0 0 1 1               (1-Bit Arithmetik)
  ---------
  0 0 1 1 1 0 0
    - 1 0 0 1 1
      ---------
      1 1 1 1 0        → muss durch XOR 0 ergeben
    - 1 0 0 1 1          ansonsten -00000 einsetzen
      ---------
        1 1 0 1 0
      - 1 0 0 1 1
        ---------
          1 0 0 1 0
        - 1 0 0 1 1
          ---------
      CRC   0 0 0 1
```

3. m Nullen ersetzen $(z^m = z^4)$ $\hat{\underline{c}} = 10000100001$

Decoder : 1. Polynomdivision

```
  1 0 0 0 0 1 0 0 0 0 1 : 1 0 0 1 1
- 1 0 0 1 1
  ---------
    1 1 1 0 0
  - 1 0 0 1 1
    ---------
      1 1 1 1 0
    - 1 0 0 1 1
      ---------
        1 1 0 1 0
      - 1 0 0 1 1
        ---------
          1 0 0 1 1
        - 1 0 0 1 1
          ---------
kein Fehler 0 0 0 0   (nur 0 = kein Fehler)
```

2. Prüfbits (CRC) entfernen $\hat{\underline{u}}$ = 100010

Fehlervektor mit 3 isolierten (nicht nebeneinander) 1-Bit Fehlern, sodass diese nicht vom Empfänger erkannt werden.

```
0 0 0 0 0 0 1 0 0 1 1   → Gleiche Länge wie Bitmuster (11-Bit) und mit 0 auffüllen
0 0 0 0 0 1 0 0 1 1 0   } 3-Mal, beliebig verschieben damit Voraussetzung erfüllt ist
0 0 1 0 0 1 1 0 0 0 0
---------------------
0 0 1 0 0 0 0 0 1 0 1   → Fehlervektor e
```