

**Dr. Jürg M. Stettbacher**

Neugutstrasse 54

CH-8600 Dübendorf

Telefon: +41 43 299 57 23

Email: dsp@stettbacher.ch

## Übung

### CRC

#### 1. Codierung und Decodierung mit CRC

In einem Kommunikationssystem können Codeworte von 16 Bit übertragen werden. Von den 16 Bits sollen 12 Bit Nutzdaten sein, der Rest soll für die CRC-Prüfsumme verwendet werden.

- (a) Wählen Sie ein geeignetes CRC-Polynom<sup>1</sup>.
- (b) Es soll das Nutzdatenwort  $\underline{u} = (B1D_h)$  übertragen werden. Bestimmen Sie das zu übertragende Codewort  $\underline{c}$ .
- (c) Machen Sie die Auswertung auf der Empfängerseite für den Fall, dass das Codewort korrekt übertragen wurde.

#### 2. Fehlerhafte Übertragung

Mit einem geeigneten CRC-Polynom vom Grad  $p$  werden mit Sicherheit alle Burstfehler der Länge  $b \leq p$  erkannt. Nehmen Sie an, im korrekten Codewort  $\underline{c} = (B1D2_h)$  mit CRC-4 sei ein Burstfehler der Länge drei aufgetreten. Verfälscht seien die Bits<sup>2</sup> 4, 3 und 2.

- (a) Berechnen Sie den Rest auf der Empfängerseite, der die fehlerhafte Übertragung anzeigt.
- (b) Erzeugen Sie für das Generatorpolynom CRC-4 einen Fehlervektor  $\underline{e}$  mit drei isolierten<sup>3</sup> 1-Bit Fehlern, so dass die Fehler vom Empfänger nicht erkannt werden können. Verifizieren sie Ihre Lösung.

#### 3. Hardware

Eine CRC-Prüfsumme soll in Hardware berechnet werden. Zur Auswahl stehen CRC-5-ITU und CRC-7.

- (a) Wählen Sie diejenige Variante, die weniger XOR-Glieder benötigt.
- (b) Zeichnen Sie das Schaltbild mit dem Schieberegister und den XOR-Gattern.
- (c) Welches ist der Rest, wenn die Hardware-Lösung das Bitmuster  $\tilde{\underline{c}} = (134_h)$  überprüft?

---

<sup>1</sup> Sie können die Tabelle mit den Generatorpolynomen aus dem Skript benutzen.

<sup>2</sup> Die Bits seien von rechts nach links nummeriert, beginnend bei 0.

<sup>3</sup> Isoliert heisst, dass die Fehlerbits im Fehlervektor nicht direkt nebeneinander stehen, sich also nicht berühren.

# 1. Codierung und Decodierung mit CRC

In einem Kommunikationssystem können Codeworte von 16 Bit übertragen werden. Von den 16 Bits sollen 12 Bit Nutzdaten sein, der Rest soll für die CRC-Prüfsumme verwendet werden.

- Wählen Sie ein geeignetes CRC-Polynom<sup>1</sup>.
- Es soll das Nutzdatenwort  $u = (B1D_h)$  übertragen werden. Bestimmen Sie das zu übertragende Codewort  $c$ .
- Machen Sie die Auswertung auf der Empfängerseite für den Fall, dass das Codewort korrekt übertragen wurde.

a) CRC-4 ( $\epsilon^4 + \epsilon + 1$ )  $\rightarrow \begin{matrix} \epsilon^4 & + \epsilon^3 & + \epsilon^2 & + \epsilon^1 & + \epsilon^0 \\ 1 & 0 & 0 & 1 & 1 \end{matrix}$

b)  $B1D_h \Rightarrow 1011 \cdot 0001 \cdot 1101_b$

$B_h = 1011$

$1_h = 0001$

$D_h = 1101$

$r = 4$

$c = 1011 \cdot 0001 \cdot 1101 \cdot 0000$

c)  $1011 \cdot 0001 \cdot 1101 \cdot 0000 : 1001 = \dots$

The diagram shows the polynomial division process. The dividend is represented as  $1011 \cdot 0001 \cdot 1101 \cdot 0000$ . The divisor is  $1001$ . The quotient is calculated as  $10011$ . The remainder is  $10$ , which is highlighted in a box at the bottom.

Der Rest wird nun an das Codewort angefügt wodurch ( $=1011 \cdot 0001 \cdot 1101 \cdot 0010$ ) & es muss nun nochmal eine Polynomdivision durch das Generatorenpolynom ( $\rightarrow 1001$ ) durchgeführt werden. Wenn Rest = 0 dann war der Empfänger das Codewort ist i.O.

## 2. Fehlerhafte Übertragung

Mit einem geeigneten CRC-Polynom vom Grad  $p$  werden mit Sicherheit alle Burstfehler der Länge  $b \leq p$  erkannt. Nehmen Sie an, im korrekten Codewort  $\underline{c} = (B1D2_h)$  mit CRC-4 sei ein Burstfehler der Länge drei aufgetreten. Verfälscht seien die Bits<sup>2</sup> 4, 3 und 2.

- (a) Berechnen Sie den Rest auf der Empfängerseite, der die fehlerhafte Übertragung anzeigt.

(b) Erzeugen Sie für das Generatorpolynom CRC-4 einen Fehlervektor  $e$  mit drei isolierten<sup>3</sup> 1-Bit Fehlern, so dass die Fehler vom Empfänger nicht erkannt werden können. Verifizieren sie Ihre Lösung.

$$c = 1011\ 0001\ 101 \cdot 0010$$

$$q) 1011 \cdot 0001 \cdot 1100 \cdot 1110 : 10011 =$$

$$\begin{array}{r}
 10011 \\
 \times 010100 \\
 \hline
 10011 \\
 \hline
 x 011111 \\
 100\ 11 \\
 \hline
 x 11\ 00\ 1 \\
 10011 \\
 \hline
 y 10100 \\
 10011 \\
 \hline
 x 011101 \\
 10011 \\
 \hline
 x 11101 \\
 10011 \\
 \hline
 x 11101 \\
 10011 \\
 \hline
 x 11100 \\
 10011 \\
 \hline
 x 1111
 \end{array}$$

$$b) C = \begin{array}{r} 1011\ 0001\ 101\ 0010 \\ 1000\ 001\ 01 \end{array}$$

$$\begin{array}{r}
 10011 \\
 10011 \\
 \hline
 110101 \\
 10011 \\
 \hline
 e = \underline{\underline{10\ 000\ 0101}}
 \end{array}$$

$$1011\ 100\ 1\ 11\ 1\ 1010 : 10011$$

$$\begin{array}{r}
 10011 \\
 \hline
 10000 \\
 10011 \\
 \hline
 11111 \\
 10011 \\
 \hline
 11001 \\
 10011 \\
 \hline
 10101 \\
 10011 \\
 \hline
 011010 \\
 10011 \\
 \hline
 10011 \\
 10011 \\
 \hline
 00
 \end{array}$$

$\Rightarrow \text{Rest} = 0$

### 3. Hardware

Eine CRC-Prüfsumme soll in Hardware berechnet werden. Zur Auswahl stehen CRC-5-ITU und CRC-7.

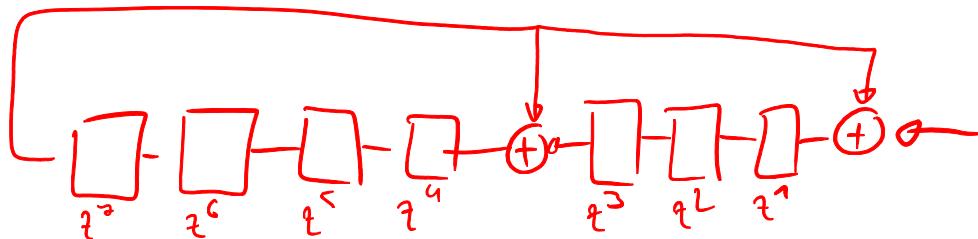
- Wählen Sie diejenige Variante, die weniger XOR-Glieder benötigt.
- Zeichnen Sie das Schaltbild mit dem Schieberegister und den XOR-Gattern.
- Welches ist der Rest, wenn die Hardware-Lösung das Bitmuster  $\tilde{c} = (134_h)$  überprüft?

$$\begin{aligned} \text{CRC-5-ITU} &= z^5 + z^4 + z^2 + 1 \\ \text{CRC-7} &= z^7 + z^3 + 1 \\ &\quad 10001001 \end{aligned}$$

Gatter: P+1

a) CRC-7

b)



c)  $134_h \Rightarrow 0001'0011'0100_s$

00000000 ← 0001'0011'0100  
 00000000 ← 100110100  
 00000001 ← 00110100  
 00000010 ← 0110100  
 00000100 ← 110100  
 00001001 ← 10100  
 00010011 ← 0100  
 00100110 ← 100  
 01001101 ← 00  
 10011010 ← 0  
10001001  
00010011  
 100110 ← "leer"

Schaltbild:

man zeichnet so viele "vierrecke" wie die Hochzahl ist -> in diesem Fall  $z^7 = 7$  Vierecke.  
bei jedem  $z^n$  kommt ein XOR-Schalter, ausser beim letzten, da geht der Schalter zurück zu den anderen XOR-Schaltern