Zyklische Codes & CRC

Revision : 257

Inhaltsverzeichnis

1 Einführung **2**

2 Grundlagen **2**

3 Erzeugung zyklischer Codes **2**

4 Verifikation **3**
4.1 Prinzip 3
4.2 Fehlererkennende Eigenschaften 4
4.2.1 Erkennung von Einfachfehlern 5
4.2.2 Erkennung von Burstfehlern 5
4.3 Fehlerkorrigierende Eigenschaften 6

5 Realisierung **6**
5.1 Bitweise Verarbeitung 6
5.2 Byteweise Verarbeitung 7

Abbildungsverzeichnis

1 Divisions-Schieberegister 6
2 LFSR mit Vormultiplikation 7

Symbolverzeichnis

CRC(x) Cyclic Redundancy Check (CRC) von x

1 Einführung

Oft besteht die Forderung den Verlust (oder auch die Manipulation) von Daten mit relativ einfachen Mitteln nachzuweisen. Einer der häufigsten Anwendungsfälle ist die Datenübertragung, bei welcher die Informationen meist blockweise strukturiert sind. Ein bevorzugtes Verfahren für diese Aufgabe ist das sogenannte CRC (Cyclic Redundancy Check), welches häufig als *Frame Check Sequence* (FCS) oder *Error Detection Code* (EDC) angewendet wird.

Das CRC-Verfahren ist eine relativ leistungsfähige (als auch einfach zu implementierende) Methode zur Fehlererkennung. Grundlage bilden sogenannte zyklische Codes (ein Spezialfall der linearen Block-Codes), welche aus theoretischer Sicht mittlerweile hinreichend untersucht sind. Dieser Beitrag wird die wesentlichen Eigenschaften solcher Codes präsentieren sowie den Zusammenhang mit dem häufig zitierten Begriff *CRC* herstellen [3, 2, 1].

2 Grundlagen

Eine grundlegende Vorbetrachtung für alle folgenden Ausführungen ist die, daß man einem beliebigen Bitstring der Länge n ein äquivalentes Polynom $P(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0$ zuordnen kann. Setzt man nun für x Elemente eines Körpers $\mathbb{F}$ (im binären Fall $\mathbb{F}_2$) ein, so erhält man ein Polynom, daß auf dem Körper $\mathbb{F}_{2^n}$ definiert ist[1]. Die Koeffizienten $a_0, \ldots, a_{n-1}$ sind ebenfalls Elemente des Körpers $\mathbb{F}_2$ und nehmen deshalb nur die Werte 0 oder 1 an[2]. Ein solches Polynom heißt irreduzibel über $\mathbb{F}_{2^n}$, wenn es sich nicht als Produkt von Polynomen kleineren Grades darstellen läßt.

3 Erzeugung zyklischer Codes

Der zu sichernde Datenblock mit einer Länge von n Bit wird zur Bildung eines zyklischen Codes als erstes um k Bit nach links verschoben. Dies bedeutet für das zugeordnete Polynom $M(x)$ eine Multiplikation mit x^k. Danach wird $M(x)$ durch ein sogenanntes Generatorpolynom $G(x) = g_k x^k + g_{k-1}x^{k-1} + \cdots + g_2x^2 + g_1x + g_0$ vom Grad k im Körper GF(2) dividiert.

$$\frac{x^k M(x)}{G(x)} = F(x) + \frac{R(x)}{G(x)} \tag{1}$$

Der dabei entstehende Divisionsrest

[1] In der Sprachweise der Mathematiker auch Galois-Feld genannt (vgl. insbesondere Anhang **??** zur Theorie).

[2] Bei Berechnungen muß im Körper $\mathbb{F}_2$ muß in diesem Zusammenhang immer berücksichtigt werden, daß statt der üblichen Addition eine modulo-2 Addition (entspricht Exklusiv-Oder) ausgeführt wird.

$$R(x) = \mathrm{CRC}\{M(x)\} = x^k M(x) \bmod G(x) \tag{2}$$

dessen Grad $k-1$ ist, wird zur Kontrolle des Informationsblockes $M(x)$ verwendet[3]. Dazu wird der entstandene CRC-Rest $R(x)$ einfach an den Datenblock $M(x)$ angehängt und so ein (gültiges) Codewort $C(x)$ des zu $G(x)$ gehörigen zyklischen Codes gebildet.

$$C(x) = x^k M(x) + R(x) \tag{3}$$

Zyklische Codes gehören zu den linearen Block-Codes. Sie haben die bemerkenswerte Eigenschaft, daß man ausgehend von einem bekannten Codewort alle anderen durch zyklische Verschiebung und Addition bestimmen kann [3]. $G(x)$ definiert alle Codeworte des zugeordneten Alphabets eindeutig und wird deshalb auch das erzeugende Polynom bzw. auch Generatorpolynom genannt.

4 Verifikation

4.1 Prinzip

Das Prinzip der Fehlererkennung besteht nun darin, daß sich jedes gültige Codewort als $C(x) = F(x)G(x)$ darstellen läßt, also ohne Rest durch $G(x)$ teilbar sein muß. Ein Nachweis dieser Eigenschaft ist recht schnell durch Kombination der Gleichungen 1 und 3 erbracht, wenn man die spezielle modulo-2 Addition $H + H = 0$ berücksichtigt.

$$\begin{aligned} C(x) &= x^k M(x) + R(x) \\ &= F(x)G(x) + R(x) + R(x) \\ &= F(x)G(x) \end{aligned}$$

Die Teilbarkeit des Codewortes $C(x)$ durch das Generatorpolynom $G(x)$ kann somit als Prüfkriterium für die Fehlerfreiheit der Übertragung verwendet werden.

$$C(x) \bmod G(x) = 0$$

Wird im Fehlerfall $C(x)$ durch ein Fehlerpolynom $E(x)$ überlagert, dann ist die Summe mit hoher Wahrscheinlichkeit nicht mehr ohne Rest durch $G(x)$ teilbar[4].

[3] $F(x)$ als ganzer Anteil, der bei der Division entsteht, ist für das Prüfverfahren nicht von Bedeutung.
[4] $C(x) + E(x)$ gehört mit hoher Wahrscheinlichkeit nicht mehr zum Codealphabet.

$$[C(x)+E(x)] \bmod G(x) \neq 0 \tag{4}$$

An dieser Stelle soll noch darauf hingeweisen sein, daß bei echten zyklischen Codes die Blocklänge $m = n + k$ nicht frei wählbar ist. Statt dessen ist sie die kleinste Zahl m, für die $(x^m + 1) \bmod G(x) = 0$ gilt, also $x^m + 1$ ohne Rest durch $G(x)$ teilbar ist. m ist also ein vom Generatorpolynom abhängiger Wert, der dann seinen Maximalwert von $m = 2^k - 1$ erreicht, wenn $G(x)$ ein nicht weiter reduzierbares (also irreduzibles) Polynom ist[5].

4.2 Fehlererkennende Eigenschaften

Da ja bekanntlich kein System ideal ist, muß man auch hier einräumen, daß es eine gewisse Anzahl nicht erkennbarer Fehler gibt – die sogenannte Restfehlerwahrscheinlichkeit. Sie entsteht dadurch, daß manchmal auch $E(x)$ ohne Rest durch $G(x)$ teilbar sein kann und so genau wieder ein gültiges Codewort entsteht.

$$\begin{aligned}[C(x)+E(x)] \bmod G(x) &= C(x) \bmod G(x) + E(x) \bmod G(x) \\ &= F(x)G(x) \bmod G(x) + E(x) \bmod G(x) \\ &= E(x) \bmod G(x)\end{aligned} \tag{5}$$

Praktisch wählt man bei der Verifikation üblicherweise das gleiche Verfahren wie auf der Senderseite, d. h. man muß noch die Vormultiplikation mit x^k berücksichtigen.

$$\left\{[C(x)+E(x)]x^k\right\} \bmod G(x) = \left[E(x)\,x^k\right] \bmod G(x)$$

Die wichtigsten fehlererkennenden Eigenschaften der (verkürzten) zyklischen Codes sind:

1. Es werden grundsätzlich alle Einfachfehler (1 Bit verändert) erkannt.
2. Es werden alle Fehler erkannt, bei denen Anfang und Ende des Störmusters nicht weiter als k Bit auseinanderliegen (Burstfehler).
3. Statistisch gesehen ist die Restfehlerwahrscheinlichkeit für Bitfehler höchstens 2^{-k}.
4. Für bestimmte Generatorpolynome $G(x)$ lassen sich außerdem noch weitere Spezialisierungen angeben:
   - Generatorpolynome $G(x)$, die durch $1 + x$ teilbar sind, erkennen grundsätzlich jeden Fehlerburst, der eine ungerade Anzahl von Bit verändert[6].

[5] Aus praktischen Gründen wird m allerdings oft kleiner gewählt – man nennt diese Codes dann verkürzte zyklische Codes. Sie besitzen die gleichen fehlererkennenden Eigenschaften wie echte zyklische Codes [2, 8.10].

[6] Schon allein diese Eigenschaft läßt einen Mathematiker gewöhnlich immer ein Generatorpolynom wählen, welches als $G(x) = (1 + x)P(x)$ darstellbar ist.

- Alle Polynome $G(x)$ erkennen bis zu einer maximalen Blocklänge m alle Zweifachfehler (für CRCs mit $k = 16$ ist m meist 32767).
- Polynome, für die beide erstgenannten Punkte zutreffen, erkennen jeden Dreifachfehler.

Bestimmte Klassen von Generatorpolynomen weisen in Bezug auf bestimmte Fehlerarten unter Umständen noch günstigere fehlererkennende Eigenschaften auf. Vor allem zwei Typen von linearen Gruppencodes seien für Interessierte noch erwähnt: die FIRE-Codes und die BOSE-CHAUDHURI-HOCQUENGHEM (BCH) Codes [2]. Mit Hilfe von FIRE-Codes ist nicht nur eine Fehlererkennung sondern sogar eine Korrektur von Fehlern bis zu einer definierten Anzahl von Bits möglich.

4.2.1 Erkennung von Einfachfehlern

Wir wollen nun den Nachweis erbringen, daß zyklische Codes jeden Einfachfehler erkennen können. Das Fehlerpolynom kann bei dieser Art von Störung (eines einzelnen Bits) als $E(x) = x^r$ modelliert werden. Es gilt also, die allgemeine Erkennbarkeitsbedingung nach Gleichung 5 auf diesen Spezialfall des Fehlerpolynoms $E(x)$ anzuwenden.

$$E(x) \bmod G(x) = x^r \bmod G(x) \tag{6}$$

Für $r < k$ ist sofort ersichtlich, daß der Divisionsrest $E(x)$ selbst ist, also $x^r \bmod G(x) = x^r$ gilt. Im anderen Fall ($r \geq k$) ist $E(x) \bmod G(x) \neq 0$ dadurch gegeben, daß es sich bei $G(x)$ um ein irreduzibles Polynom und insgesamt um einen sogenannten Restklassenkörper handelt (vgl. Anhang **??**).

4.2.2 Erkennung von Burstfehlern

Wir wollen nun untersuchen, welche Burstfehler unter Benutzung eines zyklischen Codes erkennbar sind. Das Fehlerpolynom soll dabei die Form $E(x) = x^l[e_r x^r + e_{r-1}x^{r-1} + \cdots + e_2 x^2 + e_1 x + 1]$ besitzen, d. h. l soll so gewählt sein, daß $e_0 = 1$ gilt. Die Fragestellung lautet also: Welche Polynome $E(x)$ sind im Sinne von Formel 5 ohne Rest durch $G(x)$ teilbar? Zur Beantwortung gehen wir von

$$E(x) \bmod G(x) = \left\{x^l\left[e_r x^r + e_{r-1}x^{r-1} + \cdots + e_2 x^2 + e_1 x + 1\right]\right\} \bmod G(x)$$

aus und bedenken, daß $E(x)$ ein Codewort des Alphabets sein müßte, um einen solchen Fehler nicht zu erkennen.

$$E(x) = x^l\left[e_r x^r + e_{r-1}x^{r-1} + \cdots + e_2 x^2 + e_1 x + 1\right] = F(x)G(x)$$

Eine derartige Darstellung ist aber ausgeschlossen, solange $r < k$ gilt.

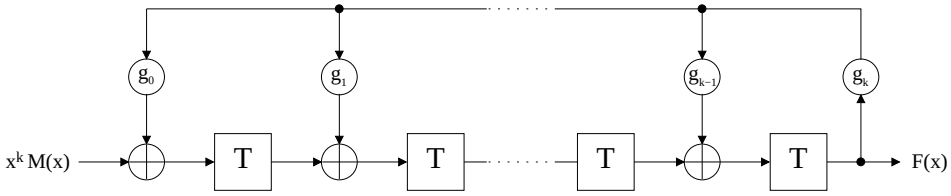

Abbildung 1: Divisions-Schieberegister

4.3 Fehlerkorrigierende Eigenschaften

Geht man davon aus, daß es sich bei $E(x)$ um einen Einfachfehler handelt, so ist mit $E(x) = x^r$ entsprechend Formel 6 das folgende CRC (auf der Empfangsseite) zu gewinnen.

$$\mathrm{CRC}\{C(x)+E(x)\} = \left\{[C(x)+E(x)]x^k\right\} \bmod G(x) = x^{r+k} \bmod G(x)$$

Je nachdem an welcher Stelle $C(x)$ nun gestört wird, ergeben sich (wegen $0 \leq r < n$) genau n mögliche Fehlerpolynome. Für alle diese $E(x)$ kann man das sogenannte Syndrom $\mathrm{CRC}[E_r(x)] = x^{r+k} \bmod G(x)$ vorab berechnen. Ein Vergleich dieser Werte mit dem aktuellen $\mathrm{CRC}\{C(x)+E(x)\} = \mathrm{CRC}\{E(x)\}$ führt zum Bitfehler, also zur Position r des gestörten Bits in $E(x) = x^r$. Sollte keine Übereinstimmung mit einem der vorberechneten Werte zu finden sein, dann muß ein Mehrfachfehler vorliegen. Solche Fehlermuster für $E(x)$ können ganz genauso wie Einzelfehler korrigiert werden[7], nur wird statt der Position r nun das gesamte Fehlerpolynom $E(x)$ von Interesse sein. Die Korrektur erfolgt in diesem Fall durch modulo-2 Addition von $E_r(x)$ zu $C(x)$, falls dessen CRC in der Syndrom-Tabelle zu finden war.

5 Realisierung

5.1 Bitweise Verarbeitung

Die bitorientierten Realisierungsvarianten sind insbesondere bei Hardware-Implementierungen sehr effizient, können aber auch softwaretechnisch emuliert werden. Die Verarbeitung erfolgt dabei mittels rückgekoppelter Schieberegister (Linear Feedback Shift Register, LFSR), dessen Inhalt nach der letzten Verschiebung den (modulo-2) Divisionsrest darstellt [4]. In Abbildung 1 ist das Grundprinzip eines solchen Divisions-Schieberegisters, welches $M(x) \bmod G(x)$ berechnet, dargestellt[8].

Die Stellung der XOR-Gatter (Symbol $\oplus$) wird durch die Koeffizienten g_i von $G(x)$, für die $g_i = 1$ gilt, bestimmt.

[7] Wobei der Maximalzahl korrigierbarer Fehler sowohl theoretische als auch praktische Grenzen gesetzt sind.

[8] Eine Bedingung für diese Realisierung ist immer $g_0 = g_k = 1$, was praktisch keine wesentliche Einschränkung darstellt.

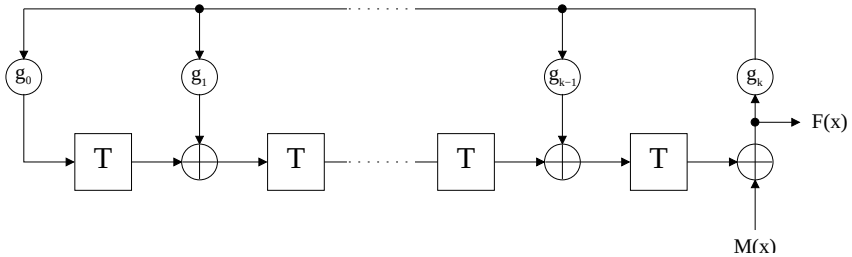

Abbildung 2: LFSR mit Vormultiplikation

Nachteil dieser Variante ist, daß die Multiplikation $x^k M(x)$, also das Nachstellen von k Null-Bits vor der Verarbeitung im LFSR notwendig ist. Eine Alternative dazu zeigt Abbildung 2, in der diese Operation durch Einspeisen von $M(x)$ an der Stelle x^k durch das Schieberegister selbst vorgenommen wird.

An dieser Stelle soll noch geklärt werden, welchen Einfluß ein bestimmter Start- bzw. Anfangsrest, auf den das LFSR in Abbildung 1 voreingestellt wird, auf den ursprünglich nach Gleichung 1 berechneten CRC-Wert hat[9]. Dazu kann der Anfangswert $R_0(x)$ als der Nachricht $M(x)$ vorangestellt aufgefaßt werden.

$$M'(x) = x^m R_0(x) + M(x)$$

Bildet man nun den CRC-Rest für die Gesamtnachricht $M'(x)$, also

$$\begin{aligned} \mathrm{CRC}\{M'(x)\} &= x^k\,[x^m R_0(x) + M(x)] \bmod G(x) \\ &= x^k x^m R_0(x) \bmod G(x) + x^k M(x) \bmod G(x) \\ &= \mathrm{CRC}\{x^m R_0(x)\} + \mathrm{CRC}\{M(x)\}, \end{aligned}$$

dann ist sofort erkennbar, daß der ohne Anfangsrest entstehende Rest $R(x) = \mathrm{CRC}\{M(x)\}$ durch Modulo-2-Addition mit $\mathrm{CRC}\{x^m R_0(x)\}$ modifiziert wird[10].

In der Praxis ist das Verfahren außerdem noch in bestimmten Modifikationen anzutreffen, z. B. Anhängen des invertierten CRC-Restes beim HDLC-Protokoll.

5.2 Byteweise Verarbeitung

Nimmt man wieder an, daß $M(x)$ die zu sichernde Nachricht ist, dann wird bei der Bildung des $\mathrm{CRC}\{M(x)\} = R(x)$ folgende Restklassendivision durchgeführt

[9] Oftmals werden alle Register auf 1 gesetzt, so daß auch wenn $M(x)$ das Nullpolynom ist, ein von Null verschiedener CRC-Rest entsteht

[10] Man kann den Fall ohne Anfangsrest ($R_0(x) = 0$) jetzt auch als Spezialfall dieser verallgemeinerten Darstellung auffassen.

$$R(x) = x^k M(x) \bmod G(x).$$

Es wird also der Polynomrest $R(x)$ so bestimmt, daß

$$x^k M(x) = F(x)G(x) + R(x)$$

unter der Nebenbedingung $R(x) < G(x)$ erfüllt ist. Nimmt man zur Nachricht $M(x)$ nun ein weiteres Byte $B(x)$ hinzu, bildet also einen neue Nachricht

$$M'(x) = x^8 M(x) + B(x),$$

dann gilt für den zugehörigen CRC-Rest $R'(x) = \mathrm{CRC}\{M'(x)\}$:

$$\begin{aligned} R'(x) &= x^k M'(x) \bmod G(x) \\ &= [x^8 x^k M(x) + x^k B(x)] \bmod G(x) \\ &= [x^8 R(x) + x^k B(x)] \bmod G(x)\,. \end{aligned} \tag{7}$$

Zerlegt man jetzt $R(x)$ so in zwei Teile $R(x) = x^{k-8}R_1(x) + R_0(x)$, daß dessen höherwertiger Teil $R_1(x)$ durch 8 Bit und der niederwertige Teil $R_0(x)$ entsprechend durch $k-8$ Bit repräsentiert wird, dann gilt

$$x^8 R_0(x) \bmod G(x) = x^8 R_0(x)$$

und Gleichung 7 vereinfacht sich zu:

$$\begin{aligned} R'(x) &= [x^k R_1(x) + x^k B(x)] \bmod G(x) + x^8 R_0(x) \\ &= x^k [R_1(x) + B(x)] \bmod G(x) + x^8 R_0(x) \\ &= \mathrm{CRC}\{R_1(x) + B(x)\} + x^8 R_0(x)\,. \end{aligned}$$

Für die durch 8 Bit darstellbaren 256 Werte ist es grundsätzlich kein Problem den CRC-Rest vorauszuberechnen und auf dieser Grundlage den Term $\mathrm{CRC}\{R_1(x) + B(x)\}$ über einen Tabellenzugriff sehr schnell zu ermitteln.

Literatur

[1] Born, Günter: *CRC-Sicherung per Software*. Toolbox, Juli 1989.

[2] Peterson, W. Wesley und E. J. Weldon, Jr.: *Error-Correcting Codes*. The MIT Press, Cambridge Massachusetts and London, England, 2. Auflage, 1972.

[3] Sweeney, P.: *Codierung zur Fehlererkennung und Fehlerkorrektur*. Carl Hanser Verlag, München, Wien, 1992.

[4] Zaks, R. und A. Lesea: *Mikroprozessor Interface Techniken*. SYBEX, 1982.

Index

Galois Field, 2

Körper, 2

Polynom, 2