RS 06 (HA) zum 30.11.2012

Paul Bienkowski, Hans Ole Hatzel

29. November 2012

1. Damit ein zyklischer Code entsteht, muss jede Bitänderung, die zum nächsten Codewort führt, irgendwann wieder umgekehrt werden. Da in einem einschrittigen Code nicht zwei Änderungen gleichzeitig durchgeführt werden dürfen, folgt daraus, dass für jedes Bit immer zwei Codewörter zu einem Paar zusammengehören. Damit ist klar, dass nur eine gerade Anzahl Codewörter möglich ist.

2. a) Der Minimalabstand beträgt 4 Bit, denn sobald sich ein Bit ändert, ändern sich damit auch die Paritätsbits der jeweiligen Spalte und Zeile, sowie das Paritätsbit der Spalten-Paritätsbits.

b) **Einbitfehler** können stets korrigiert werden, da durch die jeweiligen Zeilen- und Spaltenparitätsbits die Position in der Matrix eindeutig bestimmt werden kann. An dieser Position muss dann ein Fehler vorliegen, das entsprechende Bit muss nur invertiert werden.

Zweibitfehler werden mit Sicherheit erkannt, können jedoch nicht korrigiert werden. Es sind zwei Fälle zu unterscheiden. Solche, in denen die fehlerhaften Bits neben- oder übereinander liegen, und solche, in denen sie in unterschiedlichen Zeilen *und* Spalten liegen. Im ersten Fall wird zwar entweder die Spalte oder Zeile beider Bits erkannt, allerdings ist in der anderen Dimension das Paritätsbit zweifach invertiert worden, sodass diese Dimension nicht erkannt wird. Damit ist nicht eindeutig, wo der Fehler auftrat. Im zweiten Fall stehen dem Korrekturalgorithmus zwei Spalten und zwei Zeilen zur Verfügung, an deren 4 Kreuzungen insgesamt zwei Fehler auftraten. Die Fehler müssen immer an gegenüberliegenden Kreuzungen liegen, allerdings gibt es somit immer noch 2 Möglichkeiten, den Zweibitfehler zu lokalisieren.

Dreibitfehler werden stets als Fehler erkannt, jedoch nicht zwingend als Dreibitfehler. Die einzige Anordnung, die korrekt erkannt wird, und korrigiert werden kann, beinhaltet die drei Fehler entweder in gleicher Zeile oder gleicher Spalte. Konstellationen, bei denen jeweils zwei Fehler in der gleichen Zeile und Spalte liegen, erscheinen in den Prüfsummen wie Einbitfehler an der korrekten „Ecke", die „Korrektur" dessen bewirkt einen Vierbitfehler. Andere Konstellationen haben wie Zweibitfehler mehrere mögliche Lösungen, sodass sie nicht sicher korrigiert werden können.

c) **Vierbitfehler**, bei denen jeweils zwei Fehler in gleicher Zeile und Spalte stehen, und somit ein Rechteck aufspannen, werden nicht erkannt, da eine Spalte bzw. Zeile nur als fehlerhaft erkannt wird, wenn sie eine ungerade Anzahl Fehler enthält. In einem solchen Vierbitfehler werden somit keine fehlerhaften Spalten oder Zeilen erkannt.

Bei Fehlern in den folgenden Bits wird beispielsweise kein Fehler erkannt: $d_{0,0}, d_{0,1}, d_{3,0}, d_{3,1}$.

d) Die zwei das Rechteck aufspannende Ecken müssen in unterschiedlicher Zeile und Spalte platziert werden (Paritätsbits eingeschlossen). Da es 4 Möglichkeiten gibt, erste und zweite Ecke für das gleiche Rechteck zu definieren, muss durch 4 geteilt werden. Es ergibt sich:

$$n_4 = \frac{9 \cdot 9 \cdot 8 \cdot 8}{4} = 1296$$

Die Anzahl der Möglichkeiten, in einer 9×9 Matrix 4 verschiedene Fehler zu platzieren, ist

$$N_4 = 81 \cdot 80 \cdot 79 \cdot 78 = 39\,929\,760$$

Damit ist der Anteil der nicht erkennbaren Vierbitfehler

$$\frac{n_4}{N_4} \approx 0.00325\%$$

3. a) Das fehlerhafte Codewort lautet: 0 0 1 1 0 0 0.

$x_a = 0 \oplus 1 \oplus 0 \oplus 0 = 1$
$x_b = 0 \oplus 1 \oplus 0 \oplus 0 = 1$
$x_c = 1 \oplus 0 \oplus 0 \oplus 0 = 1$

Also ist in der Prüftabelle die Codewortstelle zu suchen, die in allen drei Prüfgruppen liegt, also c_7/d_4.

b) Für die n-te Zeile der **Generatormatrix G** liest man im Schema ab, welche Datenbits den Wert von c_n beeinflussen. Repräsentiert das Codebit c_n ein Datenbit d_m, so wird in der Matrixzeile nur das m-te Bit gesetzt. Repräsentiert das Codebit ein Prüfbit, so wird in der Matrixzeile jedes Bit gesetzt, welches zur Berechnung des Prüfbits erforderlich ist. Die Generatormatrix hat also für jedes Codebit eine Zeile, für jedes Datenbit eine Spalte.

Die **Prüfmatrix H** bildet man anhand der farbigen Punkte (Prüfgruppen). Dabei werden die Zeilen in umgekehrter Reihenfolge (von unten nach oben) in der Matrix angegeben, ein Punkt bedeutet eine 1, kein Punkt eine 0.

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & \cdots \\ 0 & 1 & 1 & 0 & 0 & 1 & \cdots \\ 0 & 0 & 0 & 1 & 1 & 1 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots \end{pmatrix}$$