Aufgabenblatt 6 Ausgabe: 02.12., Abgabe: 09.12. 12:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 6.1 (Punkte 10+10+10+10)

2D-Paritätscode: Wir betrachten den in der Vorlesung vorgestellten zweidimensionalen Paritätscode. Jeweils 49 Datenbits werden als Matrix mit 7×7 Zeilen und Spalten notiert, und zu jeder Zeile und Spalte wird ein ungerades Paritätsbit hinzugefügt. Außerdem wird noch ein weiteres Bit ganz unten rechts bestimmt, dass sich als Paritätsbit der Spalten-Paritätsbits berechnet:

$d_{0,0}$	$d_{0,1}$	$d_{0,2}$	$d_{0,3}$	$d_{0,4}$	$d_{0,5}$	$d_{0,6}$	$p_{0,7}$
$d_{1,0}$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	$d_{1,5}$	$d_{1,6}$	$p_{1,7}$
$d_{2,0}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$d_{2,4}$	$d_{2,5}$	$d_{2,6}$	$p_{2,7}$
$d_{3,0}$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	$d_{3,4}$	$d_{3,5}$	$d_{3,6}$	$p_{3,7}$
$d_{4,0}$	$d_{4,1}$	$d_{4,2}$	$d_{4,3}$	$d_{4,4}$	$d_{4,5}$	$d_{4,6}$	$p_{4,7}$
$d_{5,0}$	$d_{5,1}$	$d_{5,2}$	$d_{5,3}$	$d_{5,4}$	$d_{5,5}$	$d_{5,6}$	$p_{5,7}$
$d_{6,0}$	$d_{6,1}$	$d_{6,2}$	$d_{6,3}$	$d_{6,4}$	$d_{6,5}$	$d_{6,6}$	$p_{6,7}$
$p_{7,0}$	$p_{7,1}$	$p_{7,2}$	$p_{7,3}$	$p_{7,4}$	$p_{7,5}$	$p_{7,6}$	$p_{7,7}$

(a) Wie gross ist die Minimaldistanz d dieses Codes? Begründen Sie Ihre Antwort.

(b) Können mit diesem Code alle Einbitfehler, Zweibitfehler, und Dreibitfehler erkannt und korrigiert werden? Warum?

(c) Geben Sie ein Beispiel für einen Vierbitfehler, der vom Code nicht erkannt wird.

(d) Wie viele verschiedene Vierbitfehler der in (c) ermittelten Art gibt es? Wie gross ist der Anteil dieser Fehler in Relation zur Gesamtanzahl der möglichen Vierbitfehler?

Aufgabe 6.2 (Punkte 20+20)

Hamming-Code: Entsprechend dem in der Vorlesung vorgestellten Schema, wird ein 7-Bit Hamming-Code gebildet, um Einzelbitfehler korrigieren zu können. Wie in der Tabelle dargestellt, besitzt er vier Informationsbits (d_i) und drei Prüfbits (p_j). Insgesamt sind $2^4 = 16$ Informationen codierbar; die Codewörter sind in der linken Tabelle aufgelistet:

Nr.	c_1 p_1	c_2 p_2	c_3 d_1	c_4 p_3	c_5 d_2	c_6 d_3	c_7 d_4
0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1
2	0	1	0	1	0	1	0
3	1	0	0	0	0	1	1
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	1
6	1	1	0	0	1	1	0
7	0	0	0	1	1	1	1
8	1	1	1	0	0	0	0
9	0	0	1	1	0	0	1
10	1	0	1	1	0	1	0
11	0	1	1	0	0	1	1
12	0	1	1	1	1	0	0
13	1	0	1	0	1	0	1
14	0	0	1	0	1	1	0
15	1	1	1	1	1	1	1

Codewortstelle Bedeutung	c_1 p_1	c_2 p_2	c_3 d_1	c_4 p_3	c_5 d_2	c_6 d_3	c_7 d_4
Prüfgruppe A	*		*		*		*
Prüfgruppe B		*	*			*	*
Prüfgruppe C				*	*	*	*

Der Code ist so aufgebaut, dass die erste Prüfstelle c_1 die Informationsbits c_3, c_5 und c_7, die Prüfstelle c_2 die Stellen c_3, c_6 und c_7 und die Prufstelle c_4 die Stellen c_5, c_6 und c_7 kontrollieren. Das Prüfschema ist in der rechten Tabelle angegeben.

Um einen Einzelbitfehler zu lokalisieren, bildet man aus den Prüfgruppen A, B und C das Prüfwort mit den Stellen x_a, x_b, x_c, wobei gilt:

$$\begin{aligned} x_a &= (c_1 + c_3 + c_5 + c_7) \bmod 2 \\ x_b &= (c_2 + c_3 + c_6 + c_7) \bmod 2 \\ x_c &= (c_4 + c_5 + c_6 + c_7) \bmod 2 \end{aligned}$$

(e) Zeigen Sie, ob und wie auftretende Einzelbitfehler lokalisiert und damit korrigiert werden können. Verfälschen Sie dazu die Codewortstelle c_6 des 7. Codewortes (s. Tabelle) und bilden Sie die Prüfworte.

(f) Wie wird der Index i einer fehlerhaften Codewortstelle errechnet?

Aufgabe 6.3 (Punkte 10+10)

Kanonische Formen: Die beiden folgenden Funktionen einer 3-bit Variablen x sind in der kanonischen DNF, der kanonischen KNF und der Read-Muller-Form zu notieren.

(a) $f(x) = (\overline{x_3} \vee x_2) \wedge (\overline{x_2} \vee x_3)$

(b) $g(x) = \overline{x_3} \oplus \overline{x_2}$