Aufgabenblatt 6

Ausgabe 29/11/2010, Abgabe bis 06/12/2010 12:00

Name(n):
Matrikelnummer(n):
Übungsgruppe:

Aufgabe 6.1 Hamming-Code (20+20 Punkte)

Als *Hamming*-Codes bezeichnet man eine Klasse von Codes, die Einzelbitfehler korrigieren können. Der folgende 7-Bit Hamming-Code besitzt vier Informationsbits (I) und drei Prüfbits (P), so dass insgesamt $2^4 = 16$ Informationen codierbar sind.

(Hinweis: Die Anordnung der Daten- und Prüfbits hat auf die Funktion des Codes natürlich keinen Einfluss. Die hier gewählte Reihenfolge entspricht nicht der Variante im Skript.)

	Codewortstelle						
Nr.	a_1	a_2	a_3	a_4	a_5	a_6	a_7
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

	Codewortstelle						
Nr.	a_1	a_2	a_3	a_4	a_5	a_6	a_7
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

Der Code ist derart aufgebaut, dass die erste Prüfstelle a_1 die Informationsbits a_3, a_5 und a_7, die Prüfstelle a_2 die Stellen a_3, a_6 und a_7 und die Prüfstelle a_4 die Stellen a_5, a_6 und a_7 kontrollieren (siehe Prüfschema):

Codewortstelle	a_1	a_2	a_3	a_4	a_5	a_6	a_7
Bedeutung	P	P	I	P	I	I	I
Prüfgruppe A	x		x		x		x
Prüfgruppe B		x	x			x	x
Prüfgruppe C				x	x	x	x

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:

$$\begin{aligned}
x_a &= (a_1 + a_3 + a_5 + a_7) \bmod 2 \\
x_b &= (a_2 + a_3 + a_6 + a_7) \bmod 2 \\
x_c &= (a_4 + a_5 + a_6 + a_7) \bmod 2.
\end{aligned}$$

a) Zeigen Sie, ob und wie auftretende Einzelbitfehler lokalisiert und damit korrigiert werden können. Verfälschen Sie dazu die Codewortstelle a_5 des 9. Codewortes und bilden Sie die Prüfworte.

b) Wie wird der Index i einer fehlerhaften Codewortstelle errechnet?

Aufgabe 6.2 2D-Paritätscode (10+10+10+10 Punkte)

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. Schließlich wird noch ein weiteres Bit ganz unten rechts hinzugefügt, dass sich als Paritätsbit der Spalten-Paritätsbits berechnet.

d_{00}	d_{01}	d_{02}	d_{03}	d_{04}	d_{05}	d_{06}	p_1
d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	p_2
d_{20}	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{26}	p_3
d_{30}	d_{31}	d_{32}	d_{33}	d_{34}	d_{35}	d_{36}	p_4
d_{40}	d_{41}	d_{42}	d_{43}	d_{44}	d_{45}	d_{46}	p_5
d_{50}	d_{51}	d_{52}	d_{53}	d_{54}	d_{55}	d_{56}	p_6
d_{60}	d_{61}	d_{62}	d_{63}	d_{64}	d_{65}	d_{66}	p_7
p_{00}	p_{01}	p_{02}	p_{03}	p_{04}	p_{05}	p_{06}	p_{07}

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 6.3 Kanonische Formen (10+10 Punkte)

Folgende Funktionen sind in der kanonischen DNF, kanonischen KNF und Read-Muller-Form zu notieren:

a) $f(x, y, z) = (\overline{x} \vee y) \wedge (\overline{y} \vee z)$

b) $g(x, y, z) = \overline{x} \oplus \overline{z}$