64-041 Übung Rechnerstrukturen und Betriebssysteme

Aufgabenblatt 7 Ausgabe: 27.11., Abgabe: 04.12. 24:00

Gruppe	
Name(n)	**Matrikelnummer(n)**

Aufgabe 7.1 (Punkte 5+10)

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:

	c_1	c_2	c_3	c_4	c_5	c_6	c_7
Nr.	p_1	p_2	d_1	p_3	d_2	d_3	d_4
1	0	0	0	0	0	0	0
2	1	1	0	1	0	0	1
3	0	1	0	1	0	1	0
4	1	0	0	0	0	1	1
5	1	0	0	1	1	0	0
6	0	1	0	0	1	0	1
7	1	1	0	0	1	1	0
8	0	0	0	1	1	1	1
9	1	1	1	0	0	0	0
10	0	0	1	1	0	0	1
11	1	0	1	1	0	1	0
12	0	1	1	0	0	1	1
13	0	1	1	1	1	0	0
14	1	0	1	0	1	0	1
15	0	0	1	0	1	1	0
16	1	1	1	1	1	1	1

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

Für die Prüfstellen gilt:

$c_1 = c_3 \oplus c_5 \oplus c_7$
$c_2 = c_3 \oplus c_6 \oplus c_7$
$c_4 = c_5 \oplus c_6 \oplus c_7$

Um (einen) Einzelbitfehler zu lokalisieren, bildet man ein Prüfwort (x_a, x_b, x_c), wobei gilt:

$x_a = c_1 \oplus c_3 \oplus c_5 \oplus c_7$
$x_b = c_2 \oplus c_3 \oplus c_6 \oplus c_7$
$x_c = c_4 \oplus c_5 \oplus c_6 \oplus c_7$

(a) Zeigen Sie anhand eines Beispiels, wie ein auftretender Einzelbitfehler lokalisiert und damit korrigiert werden kann. Verfälschen Sie dazu in Codewort Nr. 9 die Stelle c_7 (s. Tabelle) und bilden Sie die zugehörigen Prüfbits.

(b) Wie kann man dann aus dem Prüfwort die fehlerhafte Codewortstelle bestimmen?

Aufgabe 7.2 (Punkte 5+5+5+5+5)

NAND als vollständige Basis: Geben Sie an, wie die folgenden boole'schen Funktionen durch geeignete Schaltungen nur aus NAND-Gattern gebildet werden können.

(a) $f_a(x_1, x_0) = x_1 \wedge x_0$ **and**

(b) $f_b(x_1, x_0) = x_1 \vee x_0$ **or**

(c) $f_c(x_1, x_0) = \overline{x_0}$ **not**

(d) $f_d(x_1, x_0) = \mathbf{0}$ (die Funktion, die immer eine 0 liefert)

(e) Zeigen Sie jetzt noch anhand eines Beispiels, dass die NAND-Verknüpfung, anders als AND und OR, nicht assoziativ ist. Schreiben Sie auch auf, wie das Assoziativgesetz für NAND aussehen würde.

Aufgabe 7.3 (Punkte 15+15)

Kanonische Formen: Die beiden folgenden Funktionen einer 3-bit Variablen $x = (x_2, x_1, x_0)$ sind in der kanonischen DNF, der kanonischen KNF und der Reed-Muller Form zu notieren.

(a) $f_a(x_2, x_1, x_0) = (\overline{x_2} \vee x_1) \wedge (x_2 \vee \overline{x_0})$

(b) $f_b(x_2, x_1, x_0) = \overline{x_2} \oplus x_0$

Aufgabe 7.4 (Punkte 5+10+5+10)

KV-Diagramme: Gegeben sei die folgende Schaltfunktion $f(x_3, x_2, x_1, x_0)$

(a) Übertragen Sie die Funktion f in ein KV-Diagramm. Verwenden Sie dabei die in der Vorlesung verwendete Anordnung der Variablen (s.u.).

(b) Bestimmen Sie aus dem KV-Diagramm die disjunktive Minimalform und die konjunktive Minimalform von f.

(c) Ersetzen Sie im KV-Diagramm zwei der Nullen durch Don't-Cares, so dass sich die disjunktive Minimalform weiter vereinfacht und bestimmen Sie diese.

(d) Wie lautet die Reed-Muller Form der ursprünglichen Funktion f.

x_3	x_2	x_1	x_0	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Variablenanordnung in den KV-Diagrammen:

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010