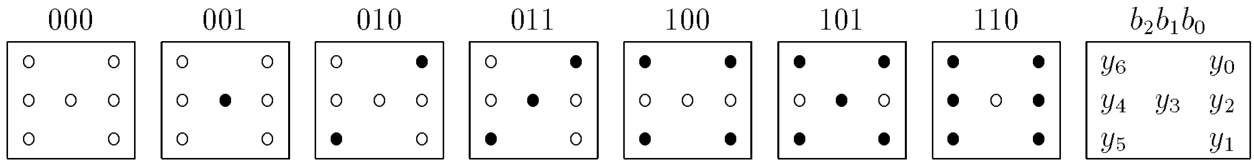
	Praxistutorium **Grundzüge der Informatik B1**	**Aufgabenblatt 1**

Aufgabe 1.1 "Würfel"-Decoder

Die folgende Aufgabe besteht in der Realisierung (Implementation und Simulation mit HADES) eines Decoders zur Umsetzung eines 3-Bit Dualcodes in einen 7-Bit n-aus-7-Code. Dabei sollen die Ziffern als Punkte eines Spielwürfels dargestellt werden. Der Dualcode 0 soll die Dunkeltastung aller Punkte des Spielwürfels bewirken.



Mit der vorgegebenen Indizierung der Punkte y_i des Spielwürfels ergeben sich die folgende Funktionstabelle, bzw. DNF's für die sieben Punkte:

b_2	b_1	b_0	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0
0	1	0	0	1	0	0	0	0	1
0	1	1	0	1	0	1	0	0	1
1	0	0	1	1	0	0	0	1	1
1	0	1	1	1	0	1	0	1	1
1	1	0	1	1	1	0	1	1	1
1	1	1	*	*	*	*	*	*	*

$$
\begin{aligned}
y_0 &= \overline{b_2}b_1\overline{b_0} \vee \overline{b_2}b_1b_0 \vee b_2\overline{b_1}\,\overline{b_0} \vee b_2\overline{b_1}b_0 \vee b_2b_1\overline{b_0}\\
y_1 &= b_2\overline{b_1}\,\overline{b_0} \vee b_2\overline{b_1}b_0 \vee b_2b_1\overline{b_0}\\
y_2 &= b_2b_1\overline{b_0}\\
y_3 &= \overline{b_2}\,\overline{b_1}b_0 \vee \overline{b_2}b_1b_0 \vee b_2\overline{b_1}b_0\\
y_4 &= b_2b_1\overline{b_0}\\
y_5 &= \overline{b_2}b_1\overline{b_0} \vee \overline{b_2}b_1b_0 \vee b_2\overline{b_1}\,\overline{b_0} \vee b_2\overline{b_1}b_0 \vee b_2b_1\overline{b_0}\\
y_6 &= b_2\overline{b_1}\,\overline{b_0} \vee b_2\overline{b_1}b_0 \vee b_2b_1\overline{b_0}
\end{aligned}
$$

a) Minimieren Sie die Booleschen Funktionen y_0 bis y_6 mit KV-Diagrammen. Verwenden Sie dabei die *don't cares* ($*$) und versuchen Sie Bündel-Minimierung zu erreichen.

b) Implementieren Sie den minimierten Decoder mit HADES und validieren Sie Ihr Ergebnis.

Aufgabe 1.2 4-Bit-Paritätscodierer

Um Bitfehler in Datenwörtern *erkennen* (nicht: korrigieren) zu können, wird häufig der sogenannte Paritätscode eingesetzt. Dabei wird an eine feste Anzahl von Datenbits (meist acht) ein Paritätsbit angehängt, so daß das Hamminggewicht des entstandenen Codewortes, also die Anzahl Eins-Bits, *gerade* bzw. *ungerade* ist; man unterscheidet somit geraden und ungeraden Paritätscode.

a) Erstellen Sie Konjunktive-/Disjunktive Normalform für einen Codierer, der aus 4-Bit-Wörtern Codewörter des geraden Paritätscodes macht, und versuchen Sie, diese zu minimieren.

b) Wieviele Konjunktions- oder Disjunktionsterme benötigt man offenbar, um n Bit mit Parität zu versehen?

Aufgabe 1.3 Logikminimierung

Um eine einfachere Logik zu erhalten, überlegen Sie sich bitte, wie sich das Paritätsbit bei Änderung genau eines Datenbits ändert. Welche logische(n) Verknüpfung(en) für zwei Operanden zeigt/zeigen dasselbe Verhalten?

a) Wählen Sie *eine* passende Verknüpfung und zeichnen Sie das Schaltnetz für den 4-Bit-Paritätscodierer.

b) Übertragen Sie dieses in HADES und testen Sie Ihre Schaltung vollständig.

Aufgabe 1.4 8-Bit-Paritätscodierer

a) Erweitern Sie Ihren Codierer auf die Verarbeitung von 8-Bit-Wörtern.

b) Sehen Sie eine zusätzliche Leitung für die Umschaltung zwischen geradem und ungeradem Paritätscode vor.

c) An einen weiteren Eingang des Schaltnetzes soll der Ausgang eines anderen Paritätscodierers angeschlossen werden können. Zeigen Sie, daß Ihr Codierer damit beliebig skalierbar wird.

Aufgabe 1.5 Wissenswertes

Die Paritätscodes mit einem Paritätsbit können offenbar genau die Hälfte aller möglichen Bitverfälschungen erkennen, nämlich die, bei denen sich eine ungerade Anzahl Bits geändert hat.

Eine häufige Art der Datenübertragung ist die bitserielle über Kupferdraht; dabei werden die Bits eines Datenwortes zeitlich nacheinander über denselben Draht gesendet. Bits können beim Empfänger falsch ankommen, falls sie in der Leitung durch elektro-magnetische Einstrahlungen verfälscht werden *oder* falls die Leitung physikalisch unterbrochen wird (durch Kabelbruch etc.). In letzterem Fall sehe der Empfänger nur Null-Bits. Meist werden je acht Datenbits mit einer Parität versehen.

a) Warum ist ein Paritätscode mit nur einem Paritätsbit in realen Anwendungen oft zur Fehlerdetektion ausreichend?

b) Warum wird man bei der geschilderten seriellen Übertragung den *ungeraden* Paritätscode dem *geraden* vorziehen?

Aufgabe 1.6 Hamming- Decoder/Encoder (optional)

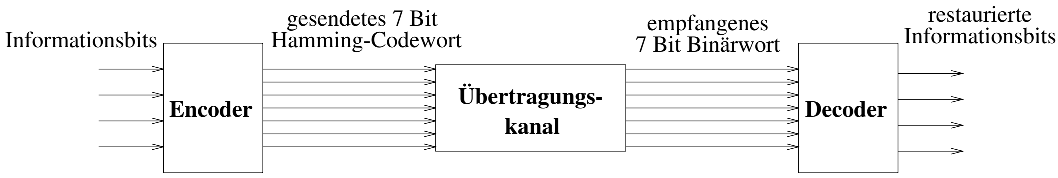
Als *Hamming*-Codes bezeichnet man eine Klasse von Codes, die Einzelbitfehler korrigieren können. Der folgende 7 Bit Hamming-Code besitzt 4 Informationsbits (I) und 3 Prüfbits (P), so daß insgesamt $2^4 = 16$ Informationen codierbar sind.

	Codewortstelle								Codewortstelle						
Nr.	a_1	a_2	a_3	a_4	a_5	a_6	a_7	Nr.	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0	0	0	0	0	0	0	0	8	1	1	1	0	0	0	0
1	1	1	0	1	0	0	1	9	0	0	1	1	0	0	1
2	0	1	0	1	0	1	0	10	1	0	1	1	0	1	0
3	1	0	0	0	0	1	1	11	0	1	1	0	0	1	1
4	1	0	0	1	1	0	0	12	0	1	1	1	1	0	0
5	0	1	0	0	1	0	1	13	1	0	1	0	1	0	1
6	1	1	0	0	1	1	0	14	0	0	1	0	1	1	0
7	0	0	0	1	1	1	1	15	1	1	1	1	1	1	1

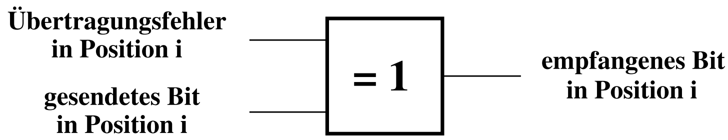
Der Code ist derart aufgebaut, daß die erste Prüfstelle a_1 die Informationsbits a_3, a_5 und a_7, die Prüfstelle a_2 die Stellen a_3, a_6 und a_7 und die Prüfstelle a_4 die Stellen a_5, a_6 und a_7 kontrollieren (siehe Prüfschema):

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

Realisieren Sie mit HADES je ein Schaltnetz zur Codierung (Encoder) und zur Decodierung (Decoder) der vier Informationsbits, so daß folgender Aufbau möglich wird:



Fügen Sie in jede Leitung des Übertragunskanals ein XOR-Gatter so ein, daß Sie einen Übertragungsfehler auf der zugehörigen Leitung simulieren können:



Simulieren Sie mit HADES die Gesamtschaltung aus Encoder, Kanal und Decoder für eine Auswahl von Einzel- und Mehrbit-Übertragungsfehlern.