Prüfung

Prof. Dr.-Ing. J. Becker

Digitaltechnik

WS 2010/11

Institut für Technik der Informationsverarbeitung, KIT

Klausur

Fr., 11.03.2011

Musterlösung

Hinweise zur Klausur

Hilfsmittel

Als Hilfsmittel zur Prüfung sind die vier Seiten vorgegebene und **ein DIN A4 Blatt** selbst geschriebene Formelsammlung zugelassen. Nicht erlaubt hingegen sind die Verwendung eines Taschenrechners, zusätzliche Unterlagen und jegliche Kommunikation mit anderen Personen.

Prüfungsdauer

Die Prüfungsdauer beträgt für die Klausur 120 Minuten.

Prüfungsunterlagen

Die Prüfungsunterlagen bestehen aus insgesamt 31 Seiten Aufgabenblättern (einschließlich diesem Titelblatt und zwei zusätzlichen Lösungsblättern). Weiterhin sind 4 zusätzliche Seiten Formelsammlung enthalten.

Bitte überprüfen Sie vor der Bearbeitung der Aufgaben auf jeder Seite oben Ihren vorgedruckten Namen und ihre Matrikelnummer.

Bei Bedarf erhalten Sie zusätzliche Lösungsblätter. Auf jedes zusätzliche Lösungsblatt ist neben dem Namen auch die Aufgabennummer mit einzutragen. Vermeiden Sie das Beschreiben der Rückseiten.

Am Ende der Prüfung sind die 31 Seiten Aufgaben- und Lösungsblätter und alle verwendeten zusätzlichen Lösungsblätter abzugeben.

Verwenden Sie zum Bearbeiten der Aufgaben lediglich dokumentenechte Schreibgeräte – keinen Bleistift sowie Rotstifte!

Aufgabe 1 Fehlererkennung und Fehlerkorrektur

Aufgabe 1.1 Fehlererkennung durch Parität

Auf einem seriellen Datenübertragungskanal treten Bitfehler auf. Deshalb soll der Kanal mittels Paritätsbits so gesichert werden, dass pro Übertragungsblock ein Bitfehler korrigierbar ist.

A) Hierzu werden jeweils 8 Bit zu einem Datenwort zusammengefasst, das um ein Paritätsbit erweitert wird. Jeweils 4 Datenworte werden anschließend zu einem Datenblock zusammengefasst, zu dem dann ein Prüfwort gebildet wird.

Ergänzen Sie den folgenden Datenblock um die jeweiligen Paritätbits für den Fall einer <u>geraden Parität</u>.

	Datenworte								Parität
	0	0	0	0	0	0	1	1	0
	1	1	1	0	0	1	0	1	1
	0	0	0	1	1	1	1	1	1
	0	1	1	0	0	1	0	1	0
Prüfwort	1	0	0	1	1	1	0	0	(0)

B) Wie groß ist die maximale Effizienz wenn bei einer anderen Blocksicherung pro 40 Bit Übertragung (gesamte Blockgröße) ein Bitfehler korrigiert werden kann? Ein Block soll dabei wieder aus vier Datenwörtern bestehen.

Anmerkung: $Effizienz = \frac{Nutzbit}{Blockgröße\ (in\ Bit)}$

*Block von 5*8 Matrix → Effizienz = 4*7/5*8 = 28/40 = 0,7 = 70%*

C) Steigt oder fällt die Effizienz mit wachsender Blockgröße? Welchen Vor- und Nachteil haben größere Blöcke?

Sie steigt …

Vorteil: Effizienzsteigerung, also mehr Datenbits pro Paritätsbit/Prüfwort

Nachteil: Der Abstand zwischen zwei erkennbaren und korrigierbaren Fehlern wächst. ODER

Der Abstand zwischen zulässigen Bündelstörungen wächst

D) Wie viele Fehler können mittels Blockparitätsprüfung pro Block in jedem Fall korrigiert werden?

Genau ein Fehler kann pro Block korrigiert werden.

E) Kann eine fehlerhafte Übertragung fälschlicherweise als korrekt ausgewertet werden? Falls JA, geben Sie ein Beispiel. Falls NEIN, begründen Sie Ihre Antwort.

Ja, Sie kann fälschlicherweise als korrekt interpretiert werden. Eine gerade Anzahl von Fehlern in jeder Spalte und Zeile (zum Beispiel 2x2) führt zu diesem Fehler.

.

Aufgabe 1.2 Fehlerkorrektur durch Hamming-Codes

Eine serielle Datenübertragung ist mit einem Hamming-Code gesichert. Jedes Codewort besteht aus 4 Datenbits (d1, d2, d3, d4) und 3 Prüfbits (p1, p2, p3).

A) Wie groß ist die Effizienz dieses Hamming-Codes? Welche Möglichkeit sehen Sie die Effizienz zu steigern?

Effizienz = 4/7 = 0,57 = 57%

Steigerung der Effizienz ist durch Verlängerung der Codewörter möglich…

⇨ Dabei immer max. # Datenbits für k Prüfbits sinnvoll

Allgemein: (2^k -1-k) / (2^k-1) für einen allgemeinen Hamming-Code

B) Wie viele Fehler können mit dem oben gegebenen Hamming-Code erkannt werden? Wie viele Fehler können mit diesem Hamming-Code korrigiert werden? Von welcher Eigenschaft hängt dies allgemein ab?

Erkennen: 2 Fehler, Korrigieren: 1 Fehler,

Eigenschaft: Hamming-Distanz 3 zwischen zwei gültigen Codewörtern. Also indirekt von der Anzahl der Prüfbits für geg. Anzahl an Datenbits

C) Vervollständigen Sie die unten stehende Tabelle, so dass ein gültiger Hamming-Code daraus abgeleitet werden kann.

lfd. Nr. **duale Kennzahl**	1 **001**	2 **010**	3 **011**	4 **100**	5 **101**	6 **110**	7 **111**
1. Stelle	p2		d3		d1		d2
2. Stelle		p1/d4	d3			d4/p1	d2
3. Stelle				p3	d1	d4/p1	d2

Für die folgende Teilaufgabe wird nun eine andere Hamming-Codierung mit 4 Datenbits (a, b, c, d) und 3 Prüfbits (y0, y1, y2), die die zugeordneten Datenbits auf <u>gerade Parität</u> ergänzen, verwendet. Der verwendete Hamming-Code ist in Abbildung 1-1 durch die Zuordnung zwischen Prüf- und Datenbits spezifiziert.

Prüfbit	Zugeordnete Datenbit
y0	a, b, d
y1	a, c, d
y2	b, c, d

Abbildung 1-1: Zweite Hamming-Codierung

D) Ein Empfänger kennt zwei unterschiedliche Übertragungsvarianten gemäß Abbildung 1-2 und Abbildung 1-3.

a	b	c	d	y2	y1	y0

Abbildung 1-2: Aufbau des Übertragungsvariante 1

a	y2	b	y1	c	y0	d

Abbildung 1-3: Aufbau des Übertragungsvariante 2

Anhand eines vorher definierten Datenstroms entscheidet der Empfänger darüber, welche Übertragungsvariante verwendet wird. Er empfängt dabei die folgende Bitsequenz: **1001101.** Vorausgesetzt die Übertragung ist fehlerfrei, welche Variante wurde vom Sender verwendet. Begründen Sie Ihre Antwort.

Nur <u>Variante zwei</u> entspricht einem gültigen Codewort. Bei Variante 1 müssten Übertragungsfehler aufgetreten sein. Z.B. wäre hier für a,b,c,d = 1 0 0 1 die Nutzbits y2,y1,y0 = 1 0 0 gültig, also y0 fehlerhaft.

Aufgabe 2 Zahlendarstellung und Konvertierung

Aufgabe 2.1 BCD

A) Addieren Sie die im Dezimalsystem gegebenen Zahlen 3468_D und 9563_D im BCD Code. Stellen Sie ihren Lösungsweg – inklusive aller notwendigen Korrekturschritte – ausführlich dar.

		BCD				Dezimalsystem
		0011	0100	0110	1000	3468
+		1001	0101	0110	0011	+9563
		-11-	1---	11--	----	
=		1100	1001	1100	1011	
Korr. wegen:.		Ps.		Ps.	Ps.	
		1100	1001	1100	1011	
+		0110		0110	0110	
		1---	--11	1--1	11--	
=	0001	0010	1010	0011	0001	
Korr. wegen:			Ps.			
		0010	1010	0011	0001	
+			0110			
		---1	11--	----	----	
=	0001	0011	0000	0011	0001	(13031)

Aufgabe 2.2 Konvertierung

A) Gegeben ist folgende Gleichung, bei der nicht alle Zahlensysteme bekannt sind:

$$\mathbf{370_H - 111_{H/O/B} = 0011\ 0010\ 0111_{BCD/B}}$$

Die möglichen Zahlensysteme der zweiten und dritten Zahl sind als Indizes angegeben. Bestimmen Sie rechnerisch um welche Zahlensysteme es sich handeln muss. Dokumentieren Sie Ihren Lösungsweg und geben Sie die korrekte Gleichung(en) mit den entsprechenden Zahlensystemen an.

Konvertierung:

$370_H \rightarrow 880_D$

$111_H \rightarrow 273_D$

$111_O \rightarrow 73_D$

$111_B \rightarrow 7_D$

$0011\ 0010\ 0111_{BCD} \rightarrow 327_D$

$0011\ 0010\ 0111_B \rightarrow 807_D$

Proberechnung:

$880_D - 273_D \neq 327_D$	$880_D - 273_D \neq 807_D$
$880_D - 73_D \neq 327_D$	$880_D - 73_D = 807_D$
$880_D - 7_D \neq 327_D$	$880_D - 7_D \neq 807_D$

Korrekte Gleichung:

$\rightarrow 370_H - 111_{\mathbf{O}} = 0011\ 0010\ 0111_{\mathbf{B}}$

Aufgabe 2.3 Fließkommazahlen

Zur Verwendung in einem Microcontroller wurde eine platzsparende Darstellung von Fließkommazahlen mit 16 Bit entwickelt. Das höchstwertige Bit stellt das Vorzeichen V dar, die acht Bits in der Mitte den Exponenten E und die niederwertigsten sieben Bits die Mantisse M (siehe Abbildung 2-1).

V	E_7	E_6	E_5	E_4	E_3	E_2	E_1	E_0	M_6	M_5	M_4	M_3	M_2	M_1	M_0

Abbildung 2-1: 16 Bit Fließkommazahlenformat

Für alle möglichen binären Belegungen ergibt sich der Dezimalwert Z aus nachstehender Formel (vgl. IEEE-Fließkommazahl):

$$Z = (-1)^V \cdot 2^{E-127} \cdot (1,M)$$

A) Stellen Sie die Zahl 0,875 als binäre 16-Bit-Fließkommazahl mit 7 Bit Mantisse und 8 Bit Exponent, entsprechend dem angegebenen Schema, dar. Geben Sie die Zwischenschritte bei der Umrechnung an.

V	E_7	E_6	E_5	E_4	E_3	E_2	E_1	E_0	M_6	M_5	M_4	M_3	M_2	M_1	M_0
0	0	1	1	1	1	1	1	0	1	1	0	0	0	0	0

$0{,}875 = 0{,}5 + 0{,}25 + 0{,}125 = 2^{-1} + 2^{-2} + 2^{-3} = 0{,}111_B$

Normierung der Mantisse: $0{,}111_B = 1{,}11_B * 2^{-1}$

Mantisse erweitern auf sieben Stellen, ohne führende Eins: $M = 1100000_B$

Exponent: -1, => $E = -1+127 = 126 = 01111110_B$

Zahl ist positiv => V = 0

⇨ Binärdarstellung: 0 01111110 1100000

B) Addieren Sie die in Aufgabenteil A) ermittelte Fließkommazahl zu der im selben Format vorliegenden Zahl 0011 1111 1011 0100. Geben Sie dabei alle Zwischenschritte an.

	V	E_7	E_6	E_5	E_4	E_3	E_2	E_1	E_0	M_6	M_5	M_4	M_3	M_2	M_1	M_0
	0	0	1	1	1	1	1	1	0	1	1	0	0	0	0	0
+	0	0	1	1	1	1	1	1	1	0	1	1	0	1	0	0
=	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0

1. Zahl: $+2^{(126-127)} * 1{,}1100000 = +2(127-127) * 0{,}11100000$
2. Zahl: $+2^{(127-127)} * 1{,}0110100$

Addition der Mantissen: $0{,}11100000_B$

$+ \quad 1{,}0110100_B$

$= \quad 10{,}01001000_B$

Normieren u. Kürzen der Mantisse: $10{,}01001000_B = 2^1 * 1{,}0010010_B$

Ergebnis: $2^{(127-127)} * 2^1 * 1{,}0010010_B = 2^{(128-127)} * 1{,}0010010_B$

⇨ V = 0, E = 10000000, M = 0010010

Aufgabe 3 Graphen und Relationen

Aufgabe 3.1 Eigenschaften von Relationen

Durch die in folgender Tabelle 3-1 dargestellte Matrix sei die Relation W α X zwischen W und X definiert. W, X sind Elemente der Menge M_1 = {a, b, c, d}.

W \ X	a	b	c	d
a	X		X	
b	X	X		X
c		X		X
d			X	

Tabelle 3-1: Matrixdarstellung der Relation I

A) Geben Sie für jede der folgenden Eigenschaften der Relation α an, ob diese erfüllt ist. Begründen sie ihre Antwort.

Reflexiv:

Die Relation ist nicht reflexiv, da die Hauptdiagonale nicht vollständig besetzt ist

Symmetrisch:

Die Relation ist nicht symmetrisch, da sie nicht spiegelsymmetrisch zur Hauptdiagonalen ist.

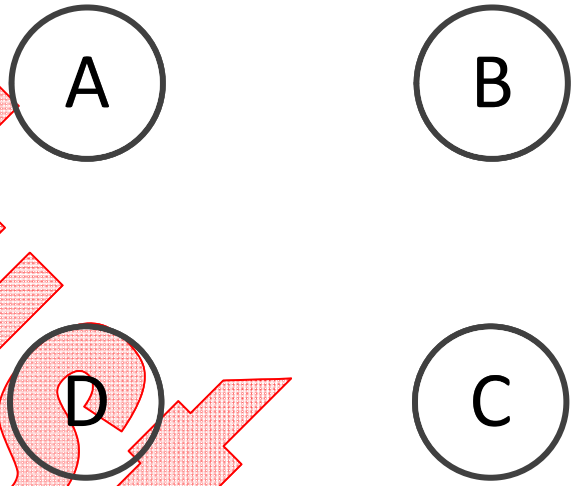
Antisymmetrisch:

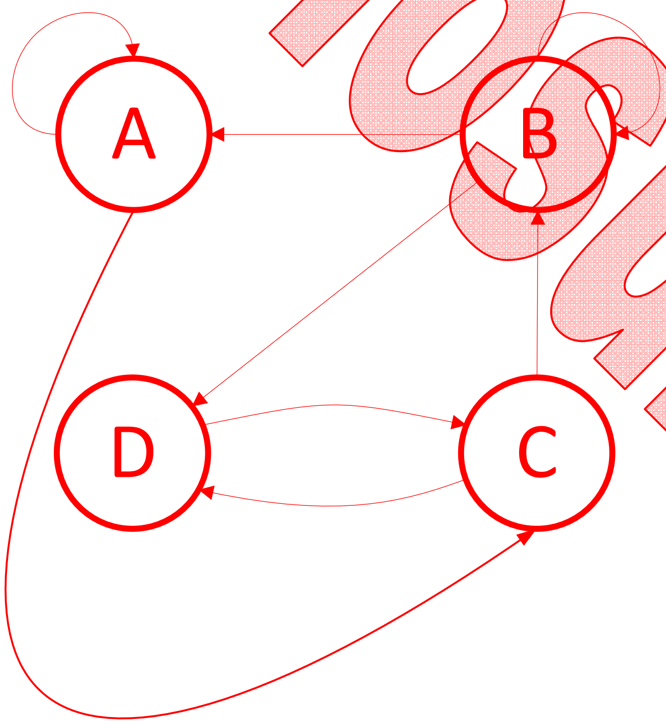
Die Relation ist nicht antisymmetrisch, da es Elemente (Y ≠Z) gibt, für die gilt $Y\alpha Z$ und $Z\alpha Y$. Z.b. c α d und d α c.

Transitiv:

Nein, die Relation ist nicht transitiv, andernfalls müsste wegen b α d und d α c auch b α c (ein gerichtete Kante von b nach c) sowie wegen a α c und c α b auch a α b (ein gerichtete Kante von a und b) erfüllt sein.

B) Zeichnen Sie den gerichteten Graphen, der die Relation α auf der Menge M_1 darstellt. Zeichnen Sie den Graph, falls möglich, als planaren Graph.





Aufgabe 3.2 Typen von Relationen

In Abbildung 3-1 ist eine weitere Relation α auf der Menge $M_2 = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ dargestellt.

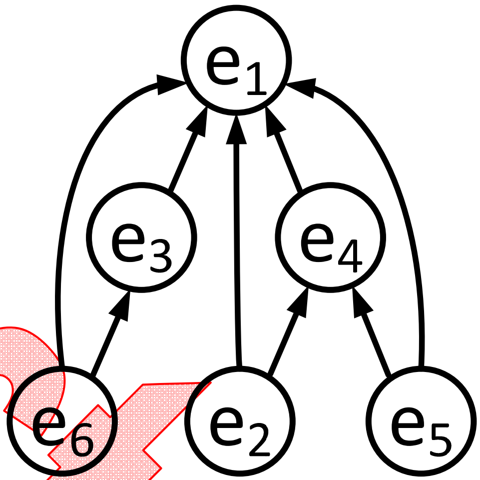


Abbildung 3-1: Graph einer Relation

A) Geben Sie die Eigenschaften dieser Relation an. Eine Begründung ist nicht nötig.

Antireflexiv / irreflexivität, antisymmetrisch, transitiv,

(asymmetrisch)

B) Handelt es sich bei der Relation α um einen bestimmten Typ von Relation? Falls ja, geben Sie diesen an.

Ja, es handelt sich um eine <u>strenge</u> Ordnungsrelation / Striktordnung.

Sie unterscheidet sich durch die Antireflexivität von der allgemeinen

Ordnungsrelation

Aufgabe 4 Boolesche Algebra und Schaltnetze

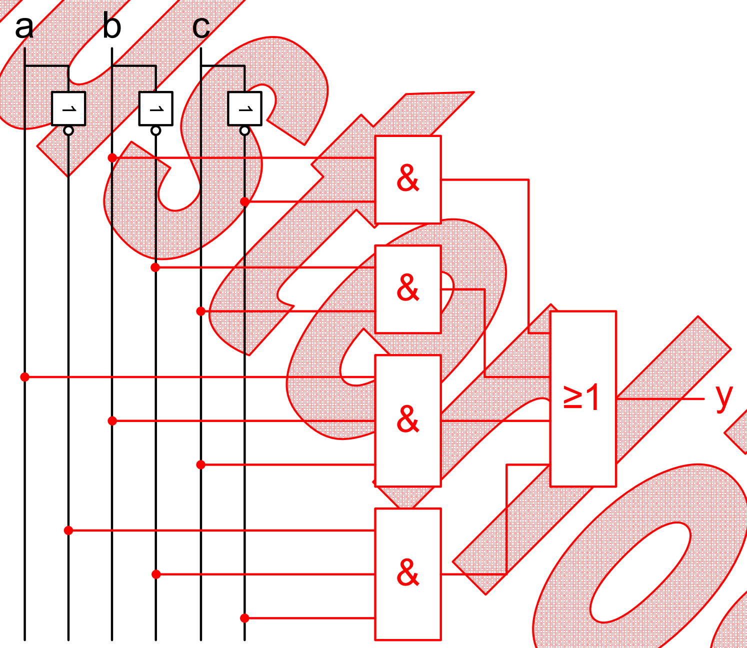
Aufgabe 4.1 Schaltnetze, Wahrheitstabelle und boolesche Algebra

Gegeben sei die folgende boolesche Funktion:

$$y_1(c, b, a) = (b \ xor \ c) \vee abc \vee \bar{a}\bar{b}\bar{c}$$

A) Zeichnen Sie ein Schaltnetz, das die Funktion y_1 realisiert. Verwenden Sie dazu nur AND-, OR- und NOT-Gatter.

$$y_1 = (b\bar{c} \vee \bar{b}c) \vee abc \vee \bar{a}\bar{b}\bar{c}$$



B) Geben Sie die Wahrheitstabelle (Tabelle 4-1) der Funktion y_1 an.

a	b	c	y_1
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Tabelle 4-1: Wahrheitstabelle der Funktion y_1

C) Geben Sie die Konjunktive Normalform (KNF) y_2 der Funktion aus der Wahrheitstabelle (Tabelle 4-1) an.

$$\textbf{KNF: } y_2(c,b,a) = (a \vee \bar{b} \vee \bar{c})\ \&\ (\bar{a} \vee b \vee c)$$

D) Zeigen Sie durch algebraische Umformung, dass die KNF y_2 aus Aufgabenteil C) dem algebraischen Ausdruck y_1 aus Teilaufgabe A) entspricht.

$y_1 = (b\bar{c} \vee \bar{b}c) \vee abc \vee \bar{a}\bar{b}\bar{c}$ // Umformung der xor-Verknüpfung

$y_1 = [(b\bar{c} \vee \bar{b}c) \& (a \vee \bar{a})] \vee abc \vee \bar{a}\bar{b}\bar{c}$ // Expansion mit fehlender Literale

$y_1 = \bar{a}b\bar{c} \vee ab\bar{c} \vee \bar{a}\bar{b}c \vee a\bar{b}c \vee abc \vee \bar{a}\bar{b}\bar{c}$ // Ausdistribuieren

$y_2 = ab \vee ac \vee \bar{a}\bar{b} \vee \bar{b}c \vee \bar{a}\bar{c} \vee b\bar{c}$ // Ausdistribuieren der beiden Maxterme

$y_2 = ab(c \vee \bar{c}) \vee a(b \vee \bar{b})c \vee \bar{a}\bar{b}(c \vee \bar{c}) \vee (a \vee \bar{a})\bar{b}c \vee \bar{a}(b \vee \bar{b})\bar{c} \vee (a \vee \bar{a})b\bar{c}$ //Erweitern

$y_2 = abc \vee ab\bar{c} \vee abc \vee a\bar{b}c \vee \bar{a}\bar{b}c \vee \bar{a}\bar{b}\bar{c} \vee a\bar{b}c \vee \bar{a}\bar{b}c \vee \bar{a}b\bar{c} \vee \bar{a}\bar{b}\bar{c} \vee ab\bar{c} \vee \bar{a}b\bar{c}$

$y_2 = abc \vee ab\bar{c} \vee a\bar{b}c \vee \bar{a}\bar{b}c \vee \bar{a}\bar{b}\bar{c} \vee \bar{a}b\bar{c}$ // Doppelte Literale streichen

$y_1 = y_2 \ => q.e.d.$

Aufgabe 4.2 Entwicklungssatz, Multiplexer- und PAL-Realisierung

A) Entwickeln Sie den Ausdruck

$$\boldsymbol{y(d,c,b,a)} = \overline{(\mathbf{a} \lor \mathbf{d})\ \&\ (\bar{\boldsymbol{a}} \lor \mathbf{c})\ \&\ (\bar{\boldsymbol{b}} \lor \mathbf{c} \lor \bar{\mathbf{d}})\ \&\ \bar{\mathbf{c}}}$$

mit Hilfe des Booleschen Entwicklungssatzes in der Reihenfolge c, a, d, b. Geben Sie alle Zwischenergebnisse an.

Hinweis: Bringen Sie den Funktionsausdruck zuerst in eine geeignete Form.

Umformung durch De Morgan:

$y(d,c,b,a) = \overline{(a \lor d)\ \&\ (\bar{a} \lor c)\ \&\ (\bar{b} \lor c \lor \bar{d})\ \&\ \bar{c}}$

$y(d,c,b,a) = \overline{(a \lor d)} \lor \overline{(\bar{a} \lor c)} \lor \overline{(\bar{b} \lor c \lor \bar{d})} \lor c$

$y(d,c,b,a) = \bar{a}\bar{d} \lor a\bar{c} \lor b\bar{c}d \lor c$

Entwicklung nach c:

$y(d,c,b,a) = c\ \&\ y(d,1,b,a) \lor \bar{c}\ \&\ y(d,0,b,a)$

$y(d,0,b,a) = \bar{a}\bar{d} \lor a \lor bd$

$y(d,1,b,a) = 1$

Entwicklung nach a:

$y(d,1,b,-) = 1$ //keine Abhängigkeit von a

$y(d,0,b,a) = a\ \&\ y(d,0,b,1) \lor \bar{a}\ \&\ y(d,0,b,0)$

$y(d,0,b,0) = \bar{d} \lor bd$

$y(d,0,b,1) = \bar{a}\bar{d} \lor 1 \lor bd = 1$

Entwicklung nach d:

Nur Abhängigkeit für c = 0 und a = 1

$y(d,0,b,a) = d\ \&\ y(1,0,b,0) \lor \bar{d}\ \&\ y(0,0,b,0)$

$y(0,0,b,0) = 1$

$y(1,0,b,0) = b$

Entwicklung nach b:

Nur Abhängigkeit für c = 0, a = 1 und d = 1

$y(1,0,b,0) = b\ \&\ y(1,0,1,0) \lor \bar{b}\ \&\ y(1,0,1,0)$

$y(1,0,0,0) = 0$

$y(1,0,1,0) = 1$

<u>Alternative Lösung:</u>

$y = \bar{a}\bar{d} \lor a\bar{c} \lor b\bar{c}d \lor c$

Entwicklung nach c:

$y = c(1) \lor \bar{c}\,(\bar{a}\bar{d} \lor a \lor bd)$

Entwicklung nach a:

$y = c(1) \lor \bar{c}\,(a(1) \lor \bar{a}(\bar{d} \lor bd))$

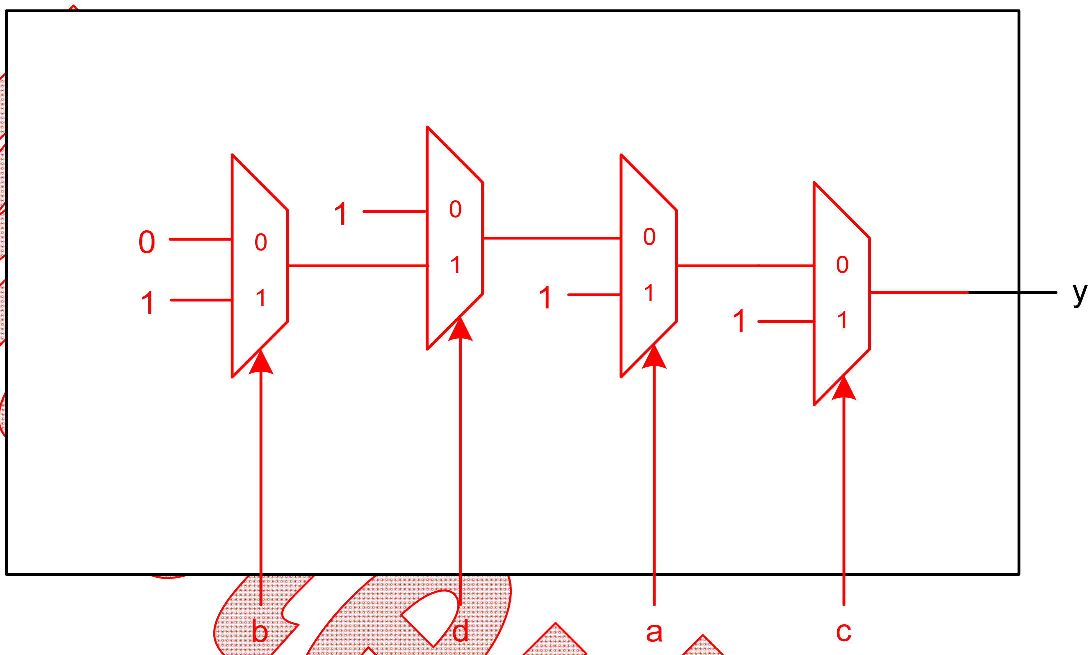
Entwicklung nach d:

$y = c(1) \lor \bar{c}\,(a(1) \lor \bar{a}(db \lor \bar{d}(1)))$

Entwicklung nach b:

$y = c(1) \lor \bar{c}\,(a(1) \lor \bar{a}(d(b(1) \lor \bar{b}(0)) \lor \bar{d}(1)))$

B) Die entwickelte Funktion aus Teilaufgabe A) soll mit 2:1 Multiplexern realisiert werden, wobei die Eingangsliterale a, b, c, d ausschließlich als Steuersignale genutzt werden sollen. Zeichnen Sie die <u>minimale</u> Multiplexerschaltung.



C) Alternativ soll die Schaltfunktion aus Teilaufgabe A) nun in einem PAL-Baustein mit einer <u>minimalen Anzahl</u> an Verknüpfungen in der UND-Matrix realisert werden. Verwenden Sie dazu Abbildung 4–1.

MUX-Realisierung zeigt: y wird nur für a=0, b=0, c=0 u. d=1 null.

⇨ KNF: $y = a \vee b \vee c \vee \overline{d}$ (ergibt minimale Anzahl an Verknüpfungen)

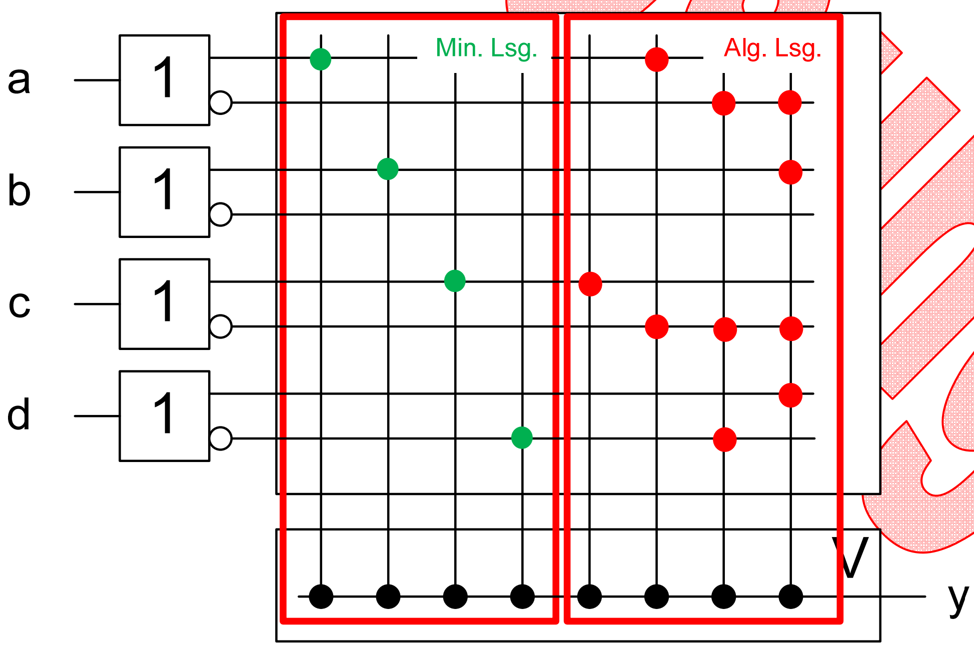


Abbildung 4–1: PAL-Schaltnetz

Aufgabe 5 Automaten und Schaltwerke

Aufgabe 5.1 Automatenentwurf

Es soll ein Automat realisiert werden um einen zyklischen Primzahlen-Zähler zu realisieren. Er soll sequenziell die ersten vier Primzahlen PZ_1 bis PZ_4 ausgeben.

Für alle Primzahlen gilt:
- Sie sind nur durch sich selbst und durch eins teilbar
- $PZ_i \in \mathbb{N}^+$ und $PZ_1 = 2$.

Als Eingabegrößen sind ein Signal „en" und ein Signal „dir" vorgesehen. Für „en = 1" wird die nächste Zahl ausgegeben, für „en = 0" wird die aktuelle Ausgabe nicht verändert. Ist „dir = 1" wird zur nächst höheren Zahl gezählt, für „dir = 0" zur nächst kleineren.

A) Geben Sie die minimal notwendige Anzahl an FlipFlops zur Zustandsspeicherung für die drei Realisierungen als Moore-, Mealy- und Medwedjew-Automat an? Begründen Sie Ihre Antwort.

4 unterschiedliche Zustände müssen kodiert werden:

*Mealy und Moore benötigen somit **ld 4=2 FlipFlops**.*

Zustände des Medwedjew Automaten kodieren gleichzeitig die Ausgabe.

Höchste auszugebende Zahl ist $7_D=111_B$.

*Somit sind **3 FlipFlops** notwendig.*

Nehmen Sie nun für die folgenden Aufgabenteile an, dass der Automat als **<u>Medwedjew-Automat</u>** realisiert werden soll.

B) Geben Sie die Zustandskodierung des Automaten an.

Zustand	*Dem Zustand zugeordnete Primzahl*	*Kodierung*
S0	2	010
S1	3	011
S2	5	101
S3	7	111

C) Zeichnen Sie das Ablaufdiagramm des beschriebenen Medwedjew Automaten. Der Automat soll nach dem Einschalten die kleinste Primzahl ausgeben.

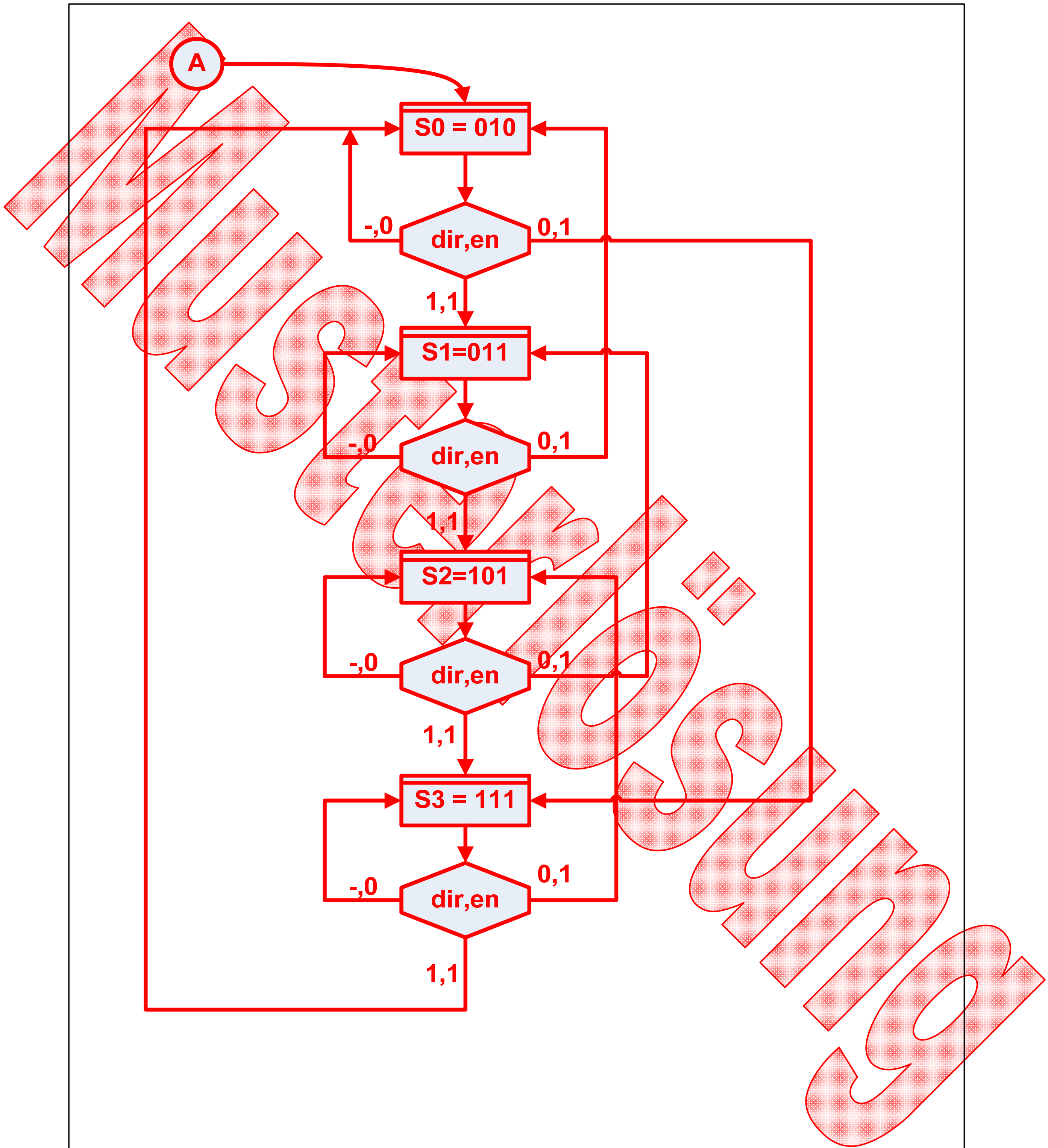


Abbildung 5–1: Ablaufdiagramm

Aufgabe 5.2 Realisierung eines Automaten

S^{ν}	$S^{\nu+1}$ E_1	E_2	E_3	E_4	A^{ν}
S0	S0	S3	S1	S3	00
S1	S0	S0	S2	S3	10
S2	S0	S1	S3	S3	01
S3	S0	S2	S0	S3	11

E_n:{e_2,e_1}
E_1:{0,0}
E_2:{0,1}
E_3:{1,0}
E_4:{1,1}

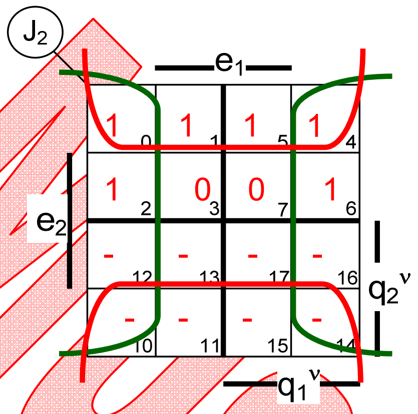
Abbildung 5–2: Automatentafel

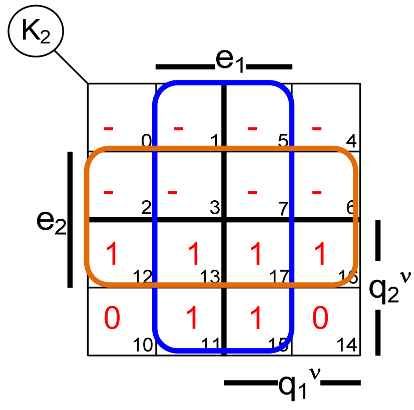
A) Wandeln Sie die Automatentafel aus Abbildung 5–2 in eine kodierte Ablauftabelle um. Vervollständigen Sie dazu die in Abbildung 5–3 gegebene Tabelle. Geben Sie in der Tabelle ebenfalls die Ansteuerung der verwendeten JK-FlipFlops an. Verwenden Sie, falls möglich, Don't-Care-Stellen zur Ansteuerung der FFs.

	Q^{ν} q_2^{ν}	q_1^{ν}	E e_2	e_1	$Q^{\nu+1}$ $q_2^{\nu+1}$	$q_1^{\nu+1}$	Y / A a_2^{ν}	a_1^{ν}	JK FF 2 J	K	JK FF 1 J	K
S0	**1**	**0**	**0**	**0**	1	0	0	0	-	0	0	-
			0	**1**	0	0			-	1	0	-
			1	**0**	0	1			-	1	1	-
			1	**1**	0	0			-	1	0	-
S1	**0**	**1**	**0**	**0**	1	0	1	0	1	-	-	1
			0	**1**	1	0			1	-	-	1
			1	**0**	1	1			1	-	-	0
			1	**1**	0	0			0	-	-	1
S2	**1**	**1**	**0**	**0**	1	0	0	1	-	0	-	1
			0	**1**	0	1			-	1	-	0
			1	**0**	0	0			-	1	-	1
			1	**1**	0	0			-	1	-	1
S3	**0**	**0**	**0**	**0**	1	0	1	1	1	-	0	-
			0	**1**	1	1			1	-	1	-
			1	**0**	1	0			1	-	0	-
			1	**1**	0	0			0	-	0	-

Abbildung 5–3: Kodierte Ablauftabelle 1

B) Erstellen Sie nun Ansteuerfunktionen des JK FlipFlops JK FF 2 zur Realisierung des Automaten. Verwenden Sie dazu die untenstehenden KV Diagramme und geben Sie Ihr Ergebnis in disjunktiver Minimalform an. Verwenden Sie zur Blockbildung auch die Freistellen.





$J_2 = \overline{e_1} + \overline{e_2}$

$K_2 = e_1 + e_2$

C) Um welchen Automatentyp handelt es sich bei dem in Aufgabenteil A) und B) realisierten Automaten? Begründen Sie Ihre Antwort.

Moore Automat, da die Ausgabe A nur vom Zustand S abhängt und nicht von der Eingabe E. Es kann sich nicht um einen Medwedjew Automat handeln, da sich der interne Zustand Q' von der Ausgabe Y/A unterscheidet

Aufgabe 6 Minimierung

Aufgabe 6.1 Primterme und DMF

Gegeben sei folgendes Symmetriediagramm der Schaltfunktion ***G***:

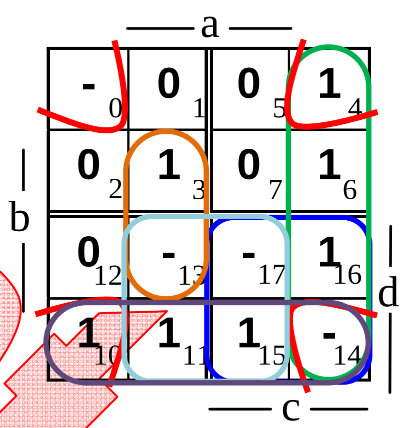


Abbildung 6-1: Symmetriediagramm

A) Geben Sie **<u>alle</u>** möglichen Primterme für eine vollständige Einsstellenüberdeckung der Funktion G aus Abbildung 6-1 an. Verwenden Sie zur Blockbildung auch die Freistellen.

$\bar{a}\bar{b}$, $\bar{a}c$, cd, ad, $\bar{b}d$, $ab\bar{c}$,

B) Geben Sie eine mögliche Überdeckung mit einer <u>minimalen</u> Anzahl an Einsblöcken in der Disjunktiven Minimalform (DMF) an. Freistellen können einbezogen werden.

$$G = ab\bar{c} \vee \bar{b}d \vee \bar{a}c$$

Aufgabe 6.2 Petrickausdruck

Ohne Streichungsregeln anzuwenden, hat ein Entwickler aus der Überdeckungstabelle einer Schaltfunktion den folgenden Petrickausdruck gebildet:

$$PA = (c \lor e)_{E1} \& (b \lor c \lor d)_{E2} \& (f)_{E3} \& (a \lor d \lor e)_{E4} \& (c)_{E5} \& (c \lor f)_{E6}$$

Um den Ausdruck nicht vollständig ausdistribuieren zu müssen, soll zunächst die Überdeckungstabelle wiedergewonnen werden.

A) Ergänzen Sie die untenstehende Überdeckungstabelle entsprechend des gegebenen Petrickausdrucks, ohne diesen zu vereinfachen. Die Präsenzvariablen a, b, c, d, e und f überdecken dabei die Einststellen E_i der Funktion.

$p_i \backslash E_i$	E_1	E_2	E_3	E_4	E_5	E_6		
a				X				
b		X						
c	X	X			X	X		
d		X		X				
e	X			X				
f			X			X		

Tabelle 6-1: Überdeckungstabelle 1

B) Bestimmen Sie alle Kernspalten aus Tabelle 6-1 und markieren Sie die entsprechende(n) Zelle(n).

Kernspalte(n):

E_3 und E_5

Aufgabe 6.3 Verfahren nach Petrick

In den folgenden Teilaufgaben sollen verschiedene Schritte des Petrick-Verfahrens durchgeführt werden.

<u>Hinweis:</u> Die folgenden Teilaufgaben können und sollen vollkommen unabhängig voneinander gelöst werden. Sie werden auch unabhängig bewertet.

A) Aus der Tabelle 6-2 wurden bereits die Kerne und die zugehörigen Spalten und Zeilen entfernt. Wenden Sie nun die <u>Spaltendominanzregel</u> auf die Überdeckungstabelle 2 an. Welche Spalte(n) können gestrichen werden? Streichen Sie die entsprechende(n) Spalte(n) und geben Sie die dominierte(n) und zugehörigen dominierende(n) Spalte(n), sowie die streichbaren Spalte(n) in der Tabelle an.

$p_i \backslash E_i$	E_1	E_3	E_4	E_6	E_7	E_8	E_9	E_{11}	E_{12}	E_{13}	E_{14}
a		X				X				X	
c	X								X		
d		X			X	X					X
f		X									
g					X			X		X	
i			X								X
j	X		X	X							X
l		X					X	X	X		
m	X			X			X		X		

Tabelle 6-2: Überdeckungstabelle 2

Dominierende Spalte(n):	E1	E3	E12	E14		
<u>Dominierte</u> Spalte(n):	E6	E8	E9	E4		
Streichbare Spalte(n):	E1	E3	E12	E14		

B) Wenden Sie nun die <u>Zeilendominanzregel</u> auf die bereits reduzierte Tabelle 6-3 an. Welche Zeile(n) können gestrichen werden? <u>Streichen Sie</u> die entsprechende(n) <u>Zeile(n) und die Spalte(n)</u> die durch dominierende Zeilen überdeckt werden. Geben Sie die dominierte(n) und zugehörigen dominierende(n) Zeile(n), sowie die streichbaren Zeile(n) in der Tabelle an. Beachten Sie dabei, dass minimale Kosten entstehen sollen. Neu entstandene Kerne dürfen mitverwendet werden.

$p_i \backslash N_i$	N_2	N_4	N_5	N_7	N_9	N_{11}	N_{12}	Kosten
a	X		X					8 GE
b		X	X			X	X	4 GE
d	X	X			X		X	7 GE
f					X		X	3 GE
g	X		X	X				6 GE
i						X		5 GE
k				X				9 GE
l		X			X			2 GE

Tabelle 6-3: Überdeckungstabelle 3

Dominierende Zeile(n):	g	g	b	l	l		
<u>Dominierte</u> Zeile(n):	k	a	i	d	f		
Streichbare Zeile(n):	k	a	i	d	f		

C) Geben Sie die Kosten der in Teilaufgabe B) gefunden Überdeckung an.

Kosten:

Verwendete Präsenzvariablen: b, g, l

Kosten: 4 GE + 6 GE + 2 GE = <u>**12 GE**</u>

Aufgabe 7 Optimale Codes

Ein Eisdielenbetreiber möchte die Eisproduktion so optimieren, dass er möglichst wenig Überschuss hat. Dazu sollen seine mobilen Verkaufswagen mit einem kleinen Datenlogger versehen werden, der die Anzahl der pro Sorte verkauften Portionen speichert und am Ende des Arbeitstages ausgelesen werden kann.

Eine per Hand geführte Zählung hat dabei die in Tabelle 7-1 dargestellte Verteilung der verschiedenen Eissorten ergeben.

Da der Speicher des Datenloggers relativ klein ist, soll zum Abspeichern der Werte eine optimale Codierung zum Einsatz kommen.

Eissorte	Birne	Erdbeere	Mango	Schokolade	Stracciatella	Vanille
Häufigkeit	6%	20%	5%	25%	23%	21%

Tabelle 7-1: Häufigkeiten der Eissortenverkäufe

Aufgabe 7.1 Huffman-Codierung

Ein Bekannter des Betreibers der Eisdiele hat dafür eine Huffman–Codierung vorgeschlagen und dafür schon einen Codebaum entworfen. Leider ist die Schrift des Bekannten aber so unleserlich, dass die Beschriftung des Baumes nicht mehr zu entziffern ist.

A) Vervollständigen Sie den Huffman-Codierbaum, indem Sie die Eissorten an den jeweils zugehörigen Blättern eintragen.

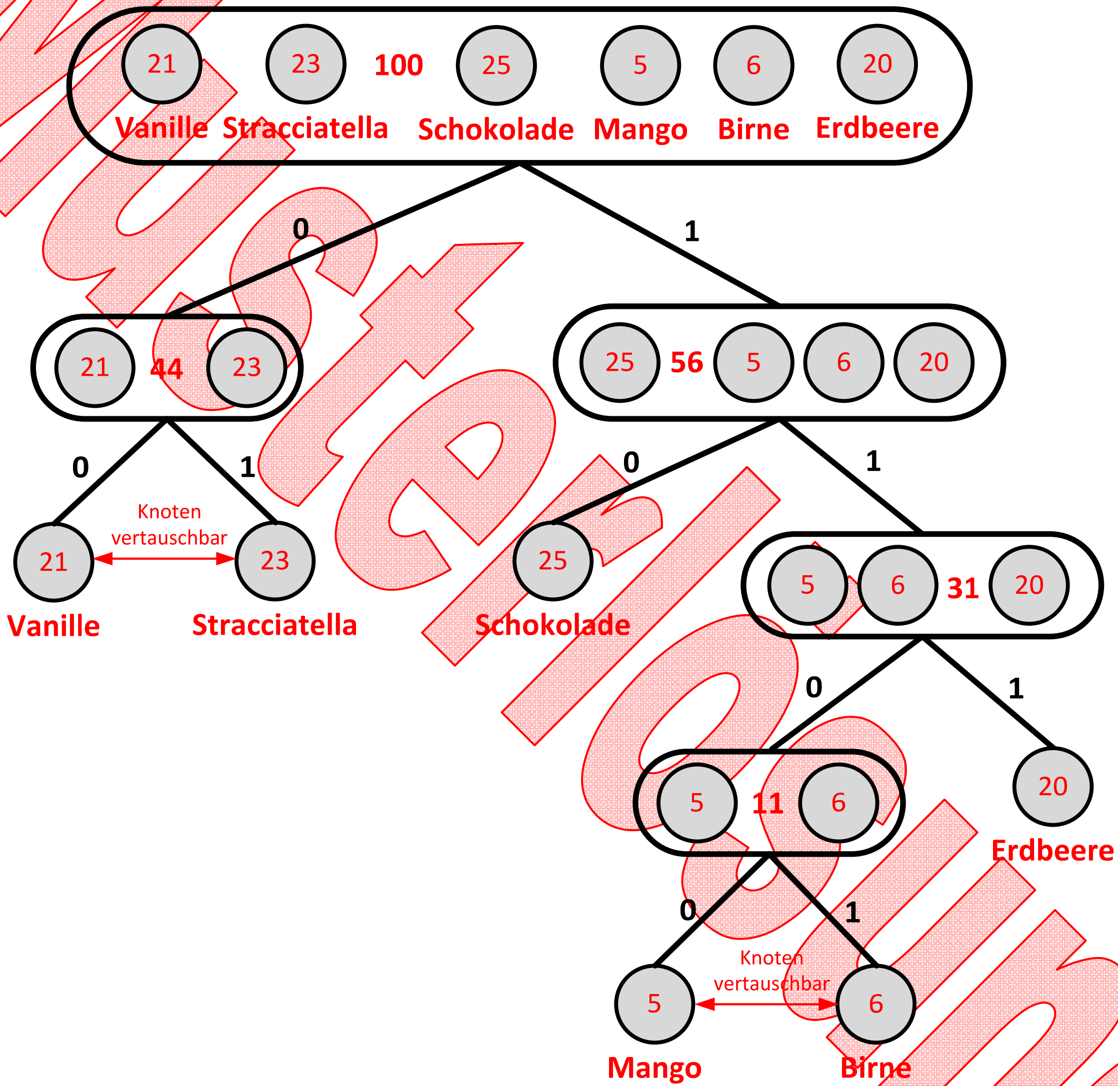


Abbildung 7-1: Huffman-Codierbaum

B) Ermitteln Sie die Codierung der einzelnen Eissorten und geben Sie diese in Tabelle 7-2 an.

Eissorte	Birne	Erdbeere	Mango	Schokolade	Stracciatella	Vanille
Codierung	1 1 0 1/0	1 1 1	1 1 0 0/1	1 0	0 1/0	0 0/1

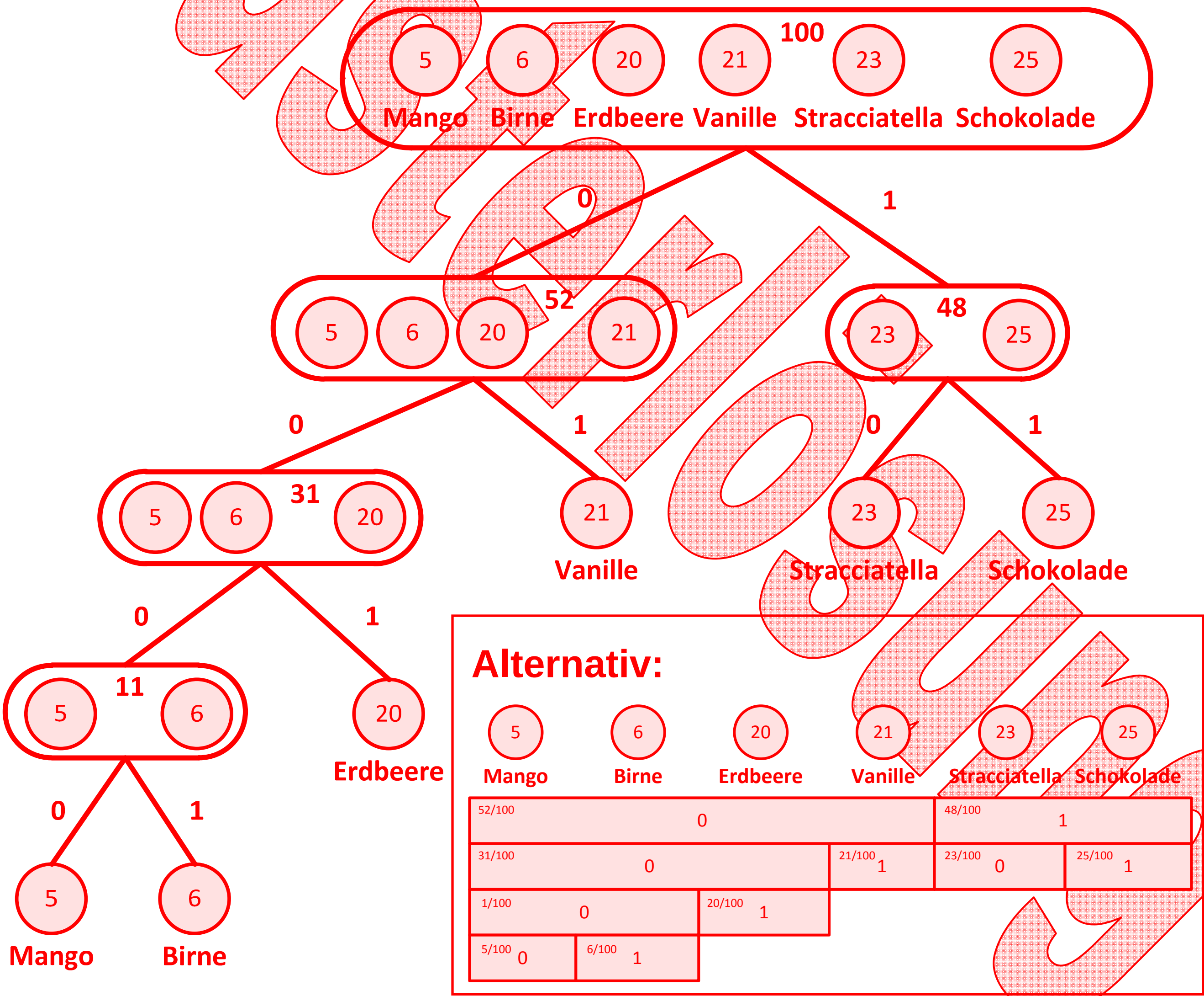
Tabelle 7-2: Huffman-Codierung der Eissorten

Aufgabe 7.2 Shannon-Fanø-Codierung

Bestimmen Sie nun die optimale Codierung nach dem Shannon-Fanø-Verfahren für die Auftrittshäufigkeiten aus Tabelle 7-1 und tragen Sie diese in Tabelle 7-3 ein.

Hinweise:

- *Sortieren Sie die Elemente zu Beginn entsprechend den Auftrittshäufigkeiten* ***aufsteigend von links nach rechts****. Falls unterschiedliche Knoten dieselbe Auftrittshäufigkeiten haben, sortieren Sie diese bitte alphabetisch.*
- *Teilen Sie eine Menge immer so auf, dass die Differenz zwischen den Summen der Auftrittshäufigen der Teilmengen minimiert wird.*
- *Verändern sie die Reihenfolge der Sortierung/Ordnung während der Anwendung des Verfahrens nicht.*
- *Weisen Sie den linken Ästen des entstehenden Baumes die „0“ zu, den rechten Ästen die „1“.*



Eissorte	Birne	Erdbeere	Mango	Schokolade	Stracciatella	Vanille
Codierung	0001	001	0000	11	10	01

Tabelle 7-3: Codierung nach Shannon-Fanø

Aufgabe 7.3 Codierungs-Auswahl

Welche der beiden Codierungen aus Aufgabe 7.1 und Aufgabe 7.2 eignet sich am besten für das konkrete Beispiel, um eine möglichst effiziente Speicherung der erfassten Daten zu erreichen? Begründen Sie Ihre Aussage!

Die Codierung mit der kleinsten mittleren Codewortlänge ermöglichst die effizienteste Speicherung der Daten.

→ Berechnung der mittleren Codewortlänge: $\bar{m} = \sum_{i=1}^{n} p(x_i) * m(x_i)$

1.1) Huffman-Code:

$$\bar{m} = 0{,}06 * 4 + 0{,}2 * 3 + 0{,}05 * 4 + 0{,}25 * 2 + 0{,}21 * 2 + 0{,}23 * 2 = 2{,}42$$

1.2) Shannon-Fano:

$$\bar{m} = 0{,}06 * 4 + 0{,}2 * 3 + 0{,}05 * 4 + 0{,}25 * 2 + 0{,}23 * 2 + 0{,}21 * 2 = 2{,}42$$

Die mittlere Codewortlänge ist bei beiden Codierungen gleich. Damit bietet keine der Codierungen einen Effizienzvorteil gegenüber der anderen Codierung!

Alternative Lösungen:

Die einzelnen Eissorten sind bei beiden Verfahren jeweils mit Codewörtern gleicher Länge codiert.

Daraus ergibt sich für beide Codierungen die gleiche mittlere Codewortlänge.

Damit bietet keine der Codierungen einen Effizienzvorteil gegenüber der anderen Codierung.

Aufgabe 8 CMOS-Schaltnetze

Aufgabe 8.1 Analyse

Gegeben sei das nicht wohldefiniert CMOS-Schaltnetz:

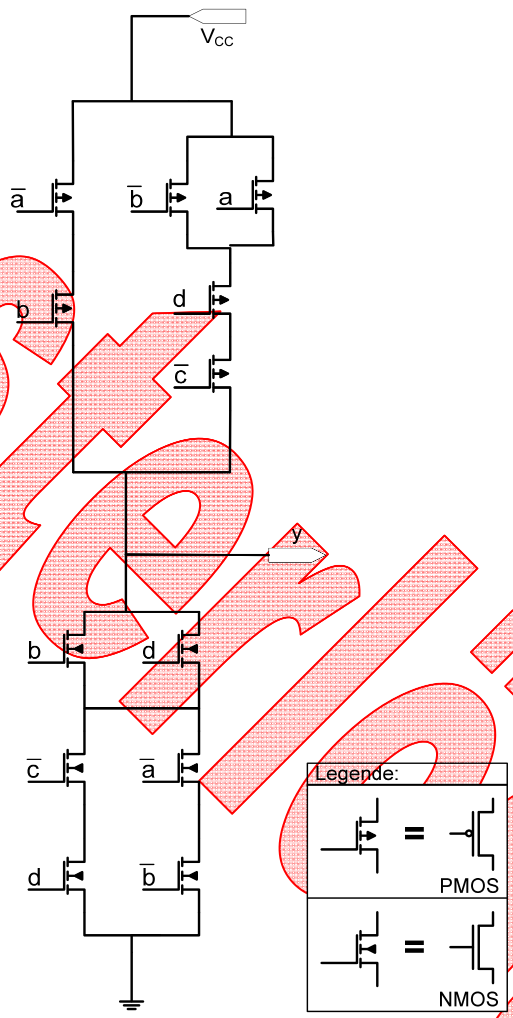


Abbildung 8-1: CMOS-Schaltnetz

Hinweis: Die Substratanschlüsse sind in der Abbildung der Übersichtlichkeit wegen nicht kontaktiert. Die Substratanschlüsse der PMOS-Transistor werden V_{cc}, die der NMOS-Transistoren mit Ground kontaktiert.

A) Geben Sie für die vorliegende Schaltung aus Abbildung 8-1 sowohl die pull-down-Funktion G als auch die pull-up-Funktion F an.

$$G = (b+d)\left(\bar{c}d + \bar{a}\bar{b}\right) = \cancel{b\bar{c}d} + \cancel{ab\bar{b}} + \bar{c}d + \bar{a}\bar{b}d = \bar{c}d + \bar{a}\bar{b}d$$

$$F = a\bar{b} + (b + \bar{a})\bar{d}c = a\bar{b} + b\bar{d}c + \bar{a}\bar{d}c$$

B) Gegeben sind folgende pull-up- bzw. pull-down-Funktionen F‘ bzw. G‘. Stellen Sie fest, ob die durch die beiden Funktionen definierte CMOS Schaltung, wohldefiniert ist. Geben Sie, falls erforderlich, die Eingangsbelegungen an, welche zu Kurzschlüssen oder einem undefinierten Ausgangssignal führen.

$$G' = \bar{a}d + \bar{c}\bar{b}d$$

$$F' = c\bar{b} + b\bar{d}a + \bar{c}\bar{d}a$$

<u>Prüfen auf Kurzschlüsse:</u>

Kurzschlussfreiheit bei: $F' \times G' = 0$

$F' \times G' = [\, \mathrm{c\bar{b}} + \mathrm{b\bar{d}a} + \mathrm{\bar{c}\bar{d}a}][\mathrm{\bar{a}d} + \mathrm{\bar{c}\bar{b}d}]$

$= \mathrm{c\bar{b}\bar{a}d} + \cancel{\mathrm{c\bar{b}\bar{c}\bar{b}d}} + \cancel{\mathrm{b\bar{d}a\bar{a}d}} + \cancel{\mathrm{b\bar{d}ac\bar{b}d}} + \cancel{\mathrm{\bar{c}\bar{d}a\bar{a}d}} + \cancel{\mathrm{\bar{c}\bar{d}ac\bar{b}d}}$

$= \mathrm{c\bar{b}\bar{a}d}$

➔ die Eingangskombination a=0, b=0, c=1, d=1 führt zu einem Kurzschluss

<u>Prüfen auf vollständige Definition:</u>

$F' + G' = [\, c\bar{b} + b\bar{d}a + \bar{c}\bar{d}a] + [\bar{a}d + \bar{c}\bar{b}d]$

Bestimme die fehlerhafte Eingangskombination durch Negation der Gleichung: $\overline{F' + G'} = 0$

$\overline{F' + G'} = \overline{c\bar{b} + b\bar{d}a + \bar{c}\bar{d}a + \bar{a}d + \bar{c}\bar{b}d}$

$= (\bar{c} + b)(\bar{b} + d + \bar{a})(c + d + \bar{a})(a + \bar{d})(c + b + \bar{d})$

$= (\bar{c}\bar{b} + \bar{c}d + \bar{c}\bar{a} + bd + b\bar{a})(ca + c\bar{d} + da + \bar{a}\bar{d})\,)(c + b + \bar{d})$

$= (\bar{c}\bar{b}da + \bar{c}\bar{b}\bar{a}\bar{d} + \bar{c}da + \bar{c}\bar{a}\bar{d} + bdca + bda + b\bar{a}c\bar{d} + b\bar{a}\bar{d})(c + b + \bar{d})$

$= bdca + bdac + b\bar{a}c\bar{d} + b\bar{a}\bar{d}c + \bar{c}dab + \bar{c}\bar{a}\bar{d}b + bdca + bda + b\bar{a}c\bar{d} + b\bar{a}\bar{d}$

$+\bar{c}\bar{b}\bar{a}\bar{d} + \bar{c}\bar{a}\bar{d} + b\bar{a}c\bar{d} + b\bar{a}\bar{d}$

$= \cancel{bdca} + \cancel{bdac} + \cancel{\bar{c}dab} + \cancel{bdca} + bda$

$+\cancel{b\bar{a}c\bar{d}} + \cancel{b\bar{a}\bar{d}c} + \cancel{\bar{c}\bar{a}\bar{d}b} + \cancel{b\bar{a}c\bar{d}} + \cancel{b\bar{a}c\bar{d}} + b\bar{a}\bar{d}$

$+\cancel{\bar{c}\bar{b}\bar{a}\bar{d}} + \bar{c}\bar{a}\bar{d}$

$= bda + b\bar{a}\bar{d} + \bar{c}\bar{a}\bar{d}$

$\overline{F' + G'} \neq 0$ ➔ G‘+F‘ ≠1 ➔ Es existieren fehlerhafte Eingangskombination:

➔ folgende Eingangskombinationen sind nicht vollständig abgedeckt:

a=1, b=1, d=1

a=0, b=1, d=0

a=0, c=0, d=0

Alternative Lösung zu Aufgabe B):

Lösung über Wahrheitstabelle:

d	c	b	a	F‘	G‘	y
0	0	0	0	0	0	Undef.
0	0	0	1	1	0	1
0	0	1	0	0	0	Undef.
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	1	1	0	0	0	Undef.
0	1	1	1	1	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	0	Undef.
1	1	0	0	1	1	KS
1	1	0	1	1	0	1
1	1	1	0	0	1	0
1	1	1	1	0	0	Undef.

Aufgabe 8.2 Realisierung

Gegeben Sei nun die pull-up Funktion eines CMOS-Schaltnetzes:

$$F'' = (\bar{a} + \bar{c}\mathrm{b})d$$

A) Geben Sie die für eine wohldefinierte CMOS-Schaltung nötige Abhängigkeit zwischen G“ und F“ an. Geben Sie die pull-down-Funktion G“ so an, dass diese als Transistorschaltnetz realisiert werden kann.

Bedingung für wohldefinierte Schaltung: $G'' = \overline{F''}$

$G'' = \overline{(\bar{a} + \bar{c}b)d} = (a(\overline{\bar{c}b})) \vee \bar{d} = (a(\bar{b} \vee c)) \vee \bar{d}$

B) Zeichnen Sie das pull-down Schaltnetz der Funktion G“ aus Aufgabenteil A).