



Prüfung

Prof. Dr.-Ing. J. Becker

Digitaltechnik

WS 2010/11

Institut für Technik der Informationsverarbeitung, KIT

Klausur

Fr., 11.03.2011

Lösungsblätter

Hinweise zur Klausur

Hilfsmittel

Als Hilfsmittel zur Prüfung sind 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 30 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 vordruckten 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 30 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	2		~12%
Aufgabe 2	Zahlendarstellung und Konvertierung	6		~13%
Aufgabe 3	Graphen und Relationen	9		~7%
Aufgabe 4	Boolesche Algebra und Schaltnetze	12		~15%
Aufgabe 5	Automaten und Schaltwerke	16		~17%
Aufgabe 6	Minimierung	20		~14%
Aufgabe 7	Optimale Codes	24		~12%
Aufgabe 8	CMOS-Schaltnetze	28		~10%
			Σ	



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 Datenwörter werden anschließend zu einem Datenblock zusammengefasst, zu dem dann ein Prüfwort gebildet wird.



Ergänzen Sie den folgenden Datenblock um die jeweiligen Paritätsbits für den Fall einer geraden Parität.

	Datenwörter								Parität
	0	0	0	0	0	0	1	1	
	1	1	1	0	0	1	0	1	
	0	0	0	1	1	1	1	1	
	0	1	1	0	0	1	0	1	
Prüfwort									

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

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



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

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

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?

- 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?

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

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

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 gerade Parität 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.

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.



Aufgabe 2.2 Konvertierung

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

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

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.

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).

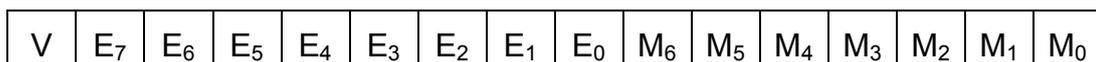


Abbildung 2-1: 16 Bit Fließkommazahlenformat

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

- 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	1	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

=

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--



Aufgabe 3 Graphen und Relationen

Aufgabe 3.1 Eigenschaften von Relationen

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

$W \setminus X$	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.



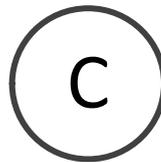
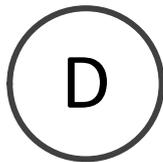
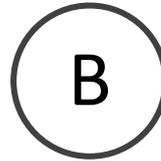
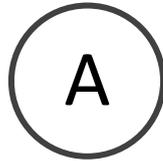
Reflexiv:

Symmetrisch:

Antisymmetrisch:

Transitiv:

- 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.

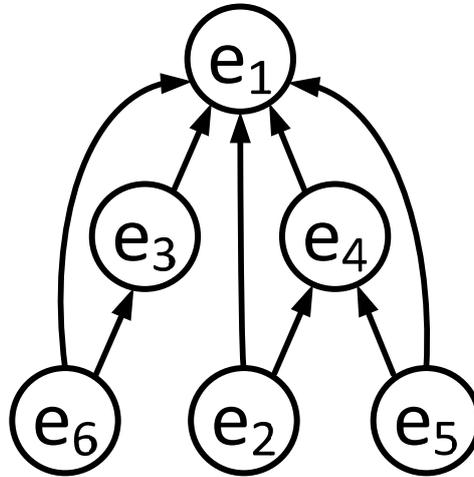


Abbildung 3-1: Graph einer Relation

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

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



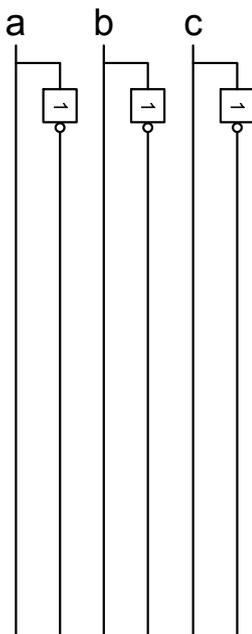
Aufgabe 4 Boolesche Algebra und Schaltnetze

Aufgabe 4.1 Schaltnetze, Wahrheitstabelle und boolesche Algebra

Gegeben sei die folgende boolesche Funktion:

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

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



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

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

$$\mathbf{KNF: } y_2(c, b, a) =$$

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

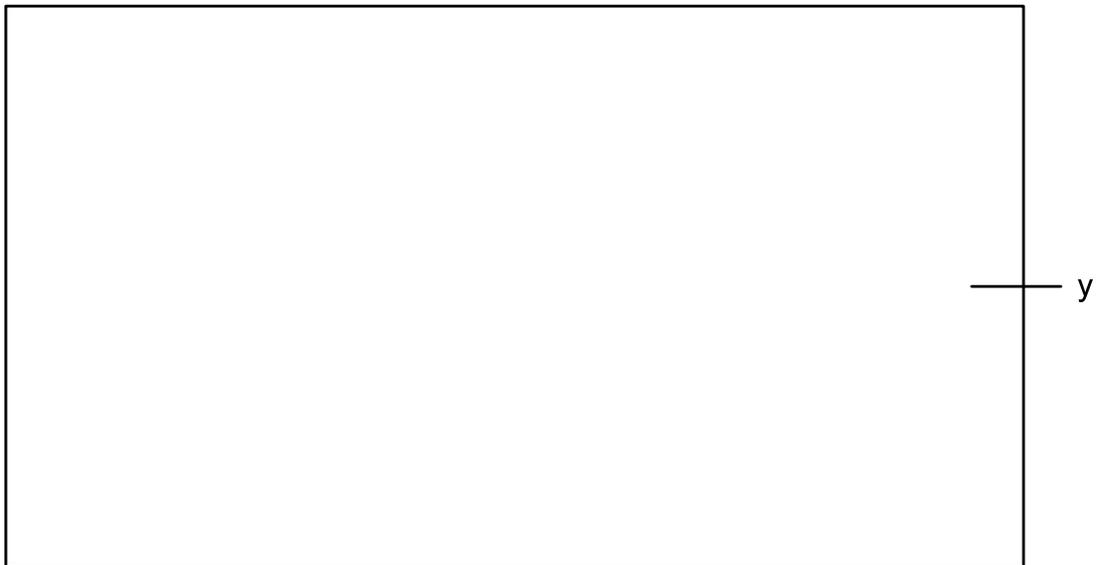
Aufgabe 4.2 Entwicklungssatz, Multiplexer- und PAL-RealisierungA) Entwickeln Sie den Ausdruck

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

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.

- 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 minimale Multiplexerschaltung.



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

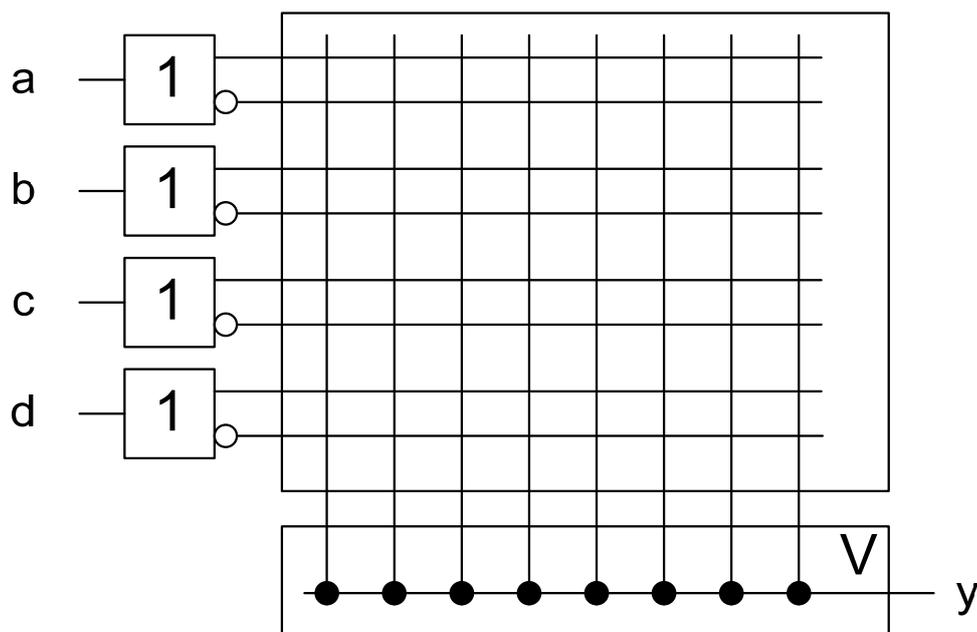


Abbildung 4–1: PAL-Schaltnetz



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.



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

B) Geben Sie die Zustandskodierung des Automaten an.



<i>Zustand</i>	<i>Dem Zustand zugeordnete Primzahl</i>	<i>Kodierung</i>
S0		
S1		
S2		
S3		

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

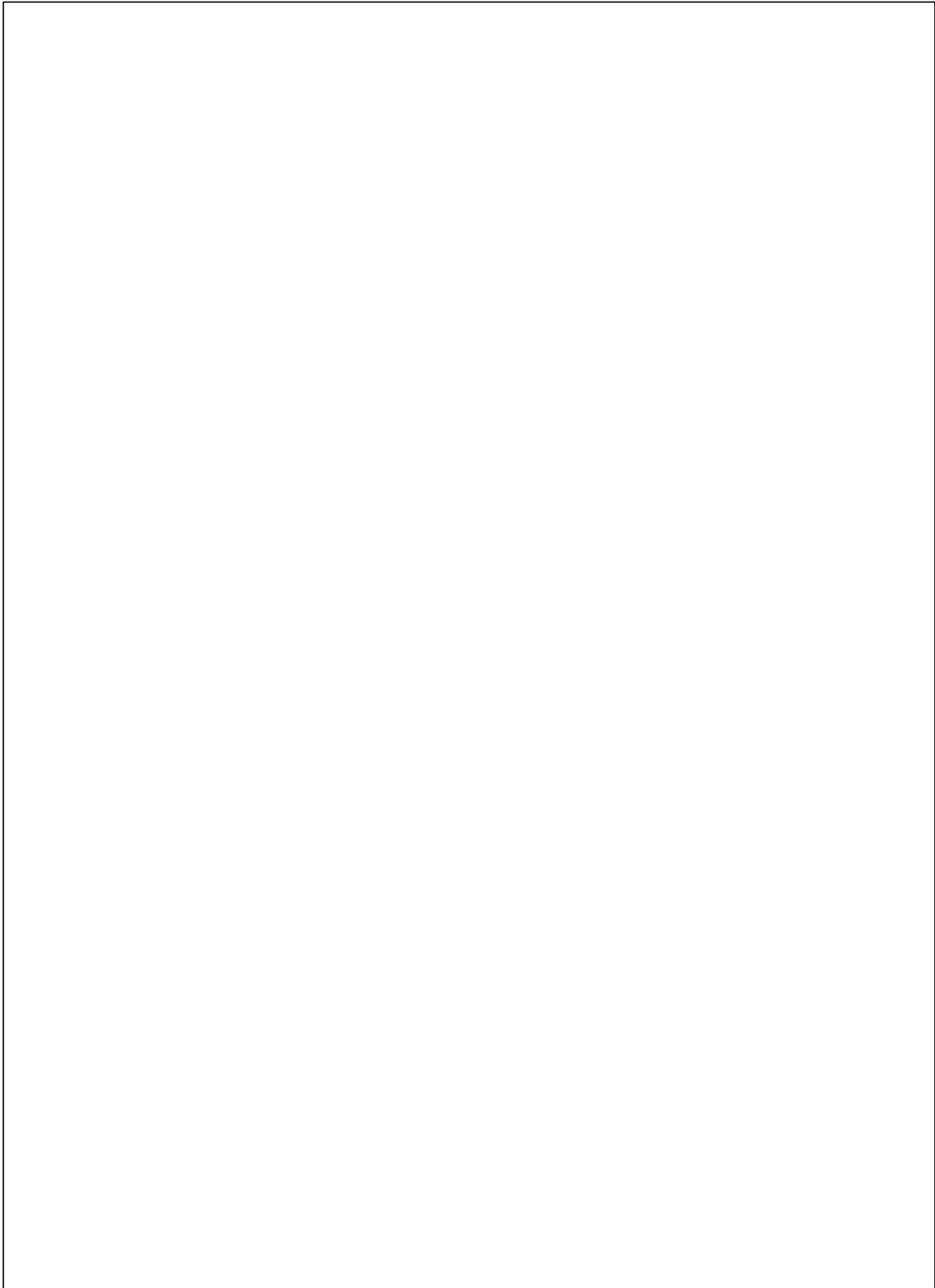


Abbildung 5–1: Ablaufdiagramm

Aufgabe 5.2 Realisierung eines Automaten

S ^v	S ^{v+1}				A ^v	E _n :{e ₂ ,e ₁ }
	E ₁	E ₂	E ₃	E ₄		
S0	S0	S3	S1	S3	00	E ₁ :{0,0}
S1	S0	S0	S2	S3	10	E ₂ :{0,1}
S2	S0	S1	S3	S3	01	E ₃ :{1,0}
S3	S0	S2	S0	S3	11	E ₄ :{1,1}

Abbildung 5–2: Automatentafel

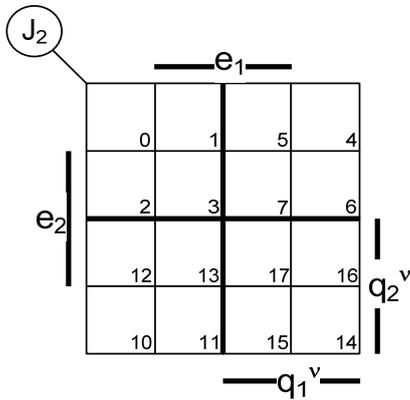
A) Wandeln Sie die Automatentafel aus Abbildung 5–2 in eine kodierte Ablauf-tabelle um. Vervollständigen Sie dazu die in Abbildung 5–3 gegebene Tabelle. Geben Sie in der Tabelle ebenfalls die Ansteuerung der verwendeten JK-FlipFlops an. Verwenden Sie, falls möglich, Don't-Care-Stellen zur Ansteuerung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



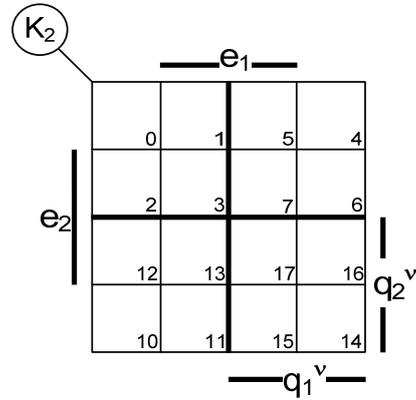
	Q ^v		E		Q ^{v+1}		Y / A		JK FF 2		JK FF 1	
	q ₂ ^v	q ₁ ^v	e ₂	e ₁	q ₂ ^{v+1}	q ₁ ^{v+1}	a ₂ ^v	a ₁ ^v	J	K	J	K
S0	1	0	0	0								
			0	1								
			1	0								
			1	1								
S1	0	1	0	0								
			0	1								
			1	0								
			1	1								
S2	1	1	0	0								
			0	1								
			1	0								
			1	1								
S3	0	0	0	0								
			0	1								
			1	0								
			1	1								

Abbildung 5–3: Kodierte Ablauf-tabelle 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.



J₂ =



K₂ =

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



Aufgabe 6 Minimierung

Aufgabe 6.1 Primterme und DMF

Gegeben sei folgendes Symmetriediagramm der Schaltfunktion G :

				— a —									
		-		0		0		5		1		4	
		0		1		0		7		1		6	
		b		-		-		-		1			
		0		12		13		17		16			
		1		10		11		15		14			
				— c —									

Abbildung 6-1: Symmetriediagramm

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



- B) Geben Sie eine mögliche Überdeckung mit einer minimalen Anzahl an Einblöcken in der Disjunktiven Minimalform (DMF) an. Freistellen können einbezogen werden.



Aufgabe 6.2 Petrickausdruck

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

$$PA = (c \vee e)_{E_1} \& (b \vee c \vee d)_{E_2} \& (f)_{E_3} \& (a \vee d \vee e)_{E_4} \& (c)_{E_5} \& (c \vee f)_{E_6}$$

Um den Ausdruck nicht vollständig ausdistribuiert 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 Einstellen E_i der Funktion.



$p_i \setminus E_i$								
a								
b								
c								
d								
e								
f								

Tabelle 6-1: Überdeckungstabelle 1

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



Kernspalte(n):

Aufgabe 6.3 Verfahren nach Petrick

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

Hinweis: 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 Spaltendominanzregel 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):						
<u>Dominierte</u> Spalte(n):						
Streichbare Spalte(n):						

- B) Wenden Sie nun die Zeilendominanzregel auf die bereits reduzierte Tabelle 6-3 an. Welche Zeile(n) können gestrichen werden? Streichen Sie die entsprechende(n) Zeile(n) und die Spalte(n) 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.



$p_i \backslash N_j$	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):							
<u>Dominierte</u> Zeile(n):							
Streichbare Zeile(n):							

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



Kosten:



Aufgabe 7 Optimale Codes

Ein Eisdienbetreiber 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.

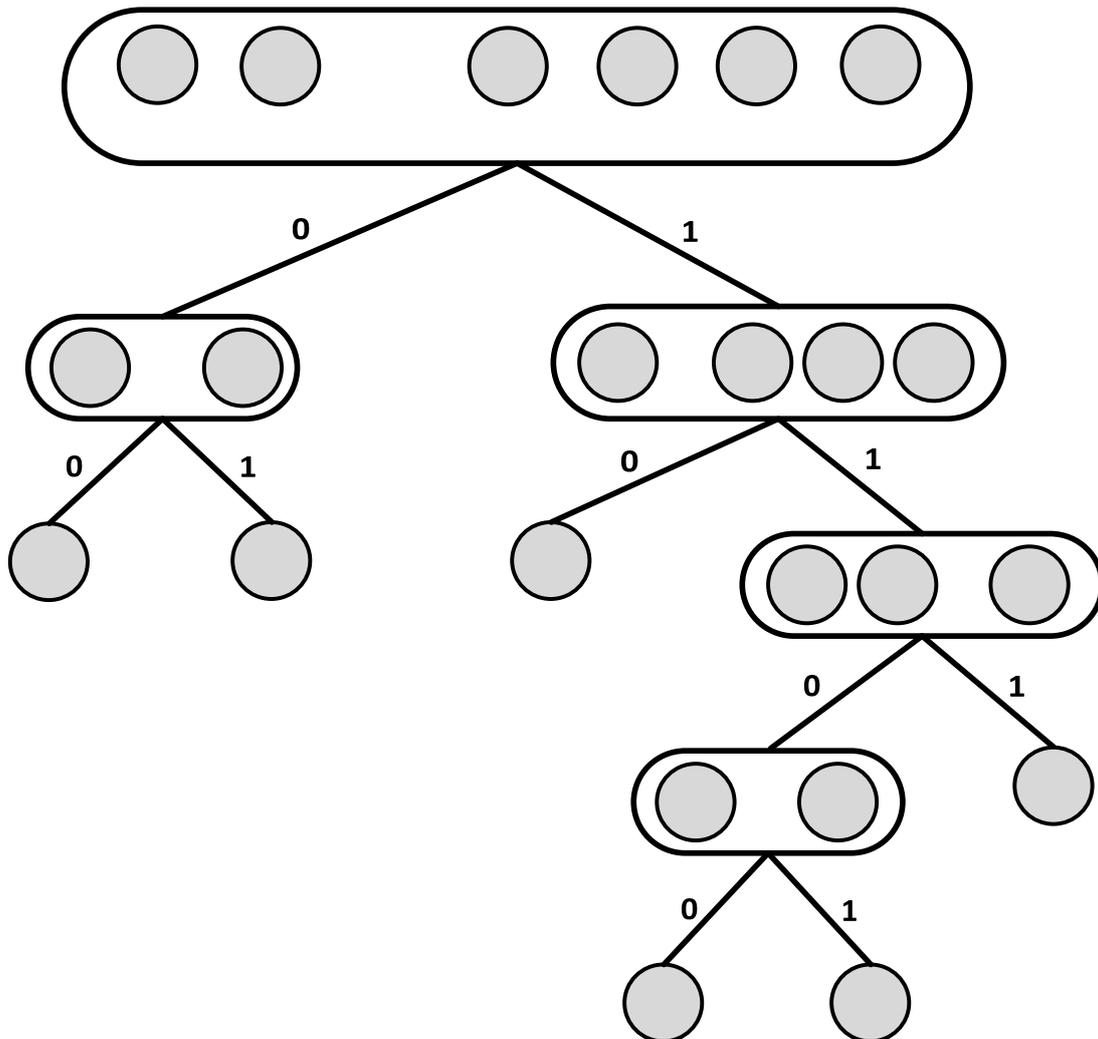


Abbildung 7-1: Huffman-Codierbaum

- 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						

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						

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!





Aufgabe 8 CMOS-Schaltnetze

Aufgabe 8.1 Analyse

Gegeben sei das nicht wohldefiniert CMOS-Schaltnetz:

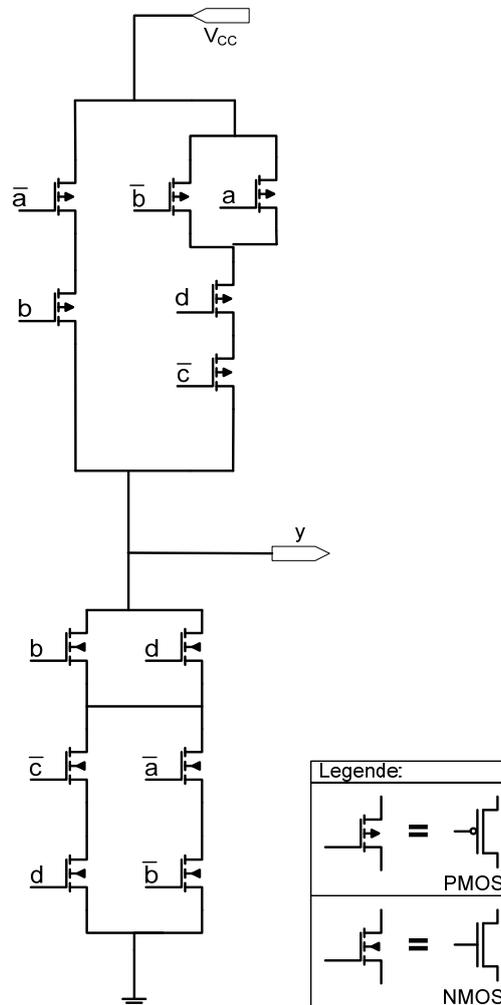


Abbildung 8-1: CMOS-Schaltnetz

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.



- 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.

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

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

Aufgabe 8.2 Realisierung

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

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

- 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.

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

