Prüfung

Prof. Dr.-Ing. J. Becker

Digitaltechnik

SS 2010

Institut für Technik der Informationsverarbeitung, KIT

Klausur

Di., 31.08.2010

Lösungsblätter

Musterlösung

Hinweise zur Klausur

Hilfsmittel

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

Prüfungsdauer

Die Prüfungsdauer beträgt für die Klausur [illegible] Minuten.

Prüfungsunterlagen

Die Prüfungsunterlagen bestehen aus insgesamt 29 Seiten Aufgabenblättern (einschließlich diesem Titelblatt und zusätzlicher Lösungsblätter). Weiterhin sind [illegible] zusätzliche Seiten Formelsammlung enthalten.

Bitte vermerken Sie vor der Bearbeitung der Aufgaben auf jeder Seite oben Ihren Namen, auf der ersten Seite zusätzlich die Matrikelnummer!

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 29 Seiten Aufgaben- und Lösungsblätter [illegible] alle verwendeten zusätzlichen Lösungsblätter abzugeben.

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

Aufgabe 1	Boolsche Algebra	5		~11%
Aufgabe 2	Minimierung	8		~14%
Aufgabe 3	Optimale Codes	12		~10%
Aufgabe 4	Mengen, Relationen und Graphen	15		~10%
Aufgabe 5	Polyadische Zahlensysteme	18		~11%
Aufgabe 6	Automaten	20		~21%
Aufgabe 7	CMOS-Schaltnetze	24		~8%
Aufgabe 8	Schaltnetze	26		~15%
			Σ	

Zusätzliches Lösungsblatt 1: (pro zusätzliches Lösungsblatt nur eine Aufgabe!)

Aufgabe ____

Zusätzliches Lösungsblatt 2: (pro zusätzliches Lösungsblatt nur eine Aufgabe!)

Aufgabe ____

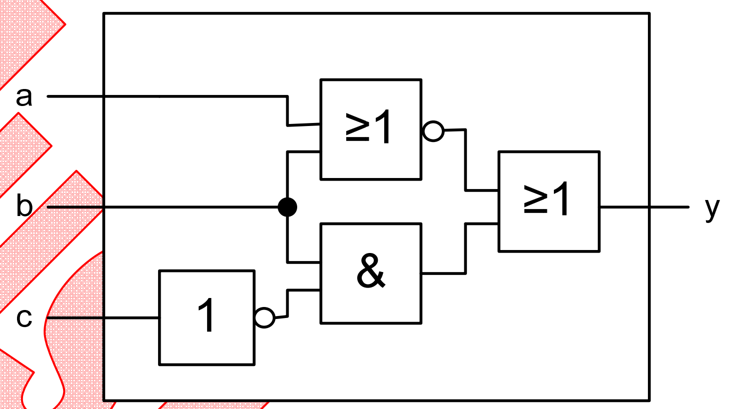
Zusätzliches Lösungsblatt 3: (pro zusätzliches Lösungsblatt nur eine Aufgabe!)

Aufgabe ____

Aufgabe 1 Boolsche Algebra

Aufgabe 1.1 Schaltnetze

Gegeben sei das folgende Schaltnetz:



A) Erstellen Sie zu dem gegebenen Schaltnetz eine entsprechende Wahrheitstabelle mit geeigneten Zwischenwerten. Verwenden Sie Tabelle 1-1.

a	b	c	$\overline{a \vee b}$	$\bar{c} \wedge b$	y
1	1	1	0	0	0
1	1	0	0	1	1
1	0	1	0	0	0
1	0	0	0	0	0
0	1	1	0	0	0
0	1	0	0	1	1
0	0	1	1	0	1
0	0	0	1	0	1

Tabelle 1-1: Wahrheitstabelle

B) Geben Sie die Disjunktive Normalform (DNF) für den Wert von y an:

DNF: $y(a,b,c) = ab\bar{c} \vee \bar{a}b\bar{c} \vee \bar{a}\bar{b}c \vee \bar{a}\bar{b}\bar{c}$

C) Entspricht der gefundene DNF-Ausdruck aus Teilaufgabe B) dem unten angegebenen Ausdruck? Begründen Sie Ihre Aussage durch eine Rechnung.

$$\overline{(a \vee b) \wedge \left(\bar{b} \vee c\right)} =$$

Gegebene Gleichung umformen:

$$\overline{(a \vee b) \wedge \left(\bar{b} \vee c\right)} = \overline{(a \vee b)} \vee \overline{(\bar{b} \vee c)} = \overline{(a \vee b)} \vee (b \wedge \bar{c}) =$$

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

DNF auf gleiche Form bringen:

$$ab\bar{c} \vee \bar{a}b\bar{c} \vee \bar{a}\bar{b}c \vee \bar{a}\bar{b}\bar{c} = b\bar{c} \vee \bar{a}\bar{b} = \bar{a}\bar{b} \vee b\bar{c}$$ q. e. d.

→ *Die Gleichungen entsprechen sich!*

Aufgabe 1.2 Entwicklungssatz

A) Entwickeln Sie den Ausdruck

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

mit Hilfe des Booleschen Entwicklungssatzes in der Reihenfolge b, a, c.

Geben Sie alle Zwischenergebnisse an.

Entwicklung nach b:

$y(c, 0, a) = \bar{a}$

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

Entwicklung nach a:

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

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

$y(c, 1,1) = \bar{c}$

$y(c, 1,0) = \bar{c}$

Entwicklung nach c:

⇨ Für b = 0 ergibt sich keine Abhängigkeit von c

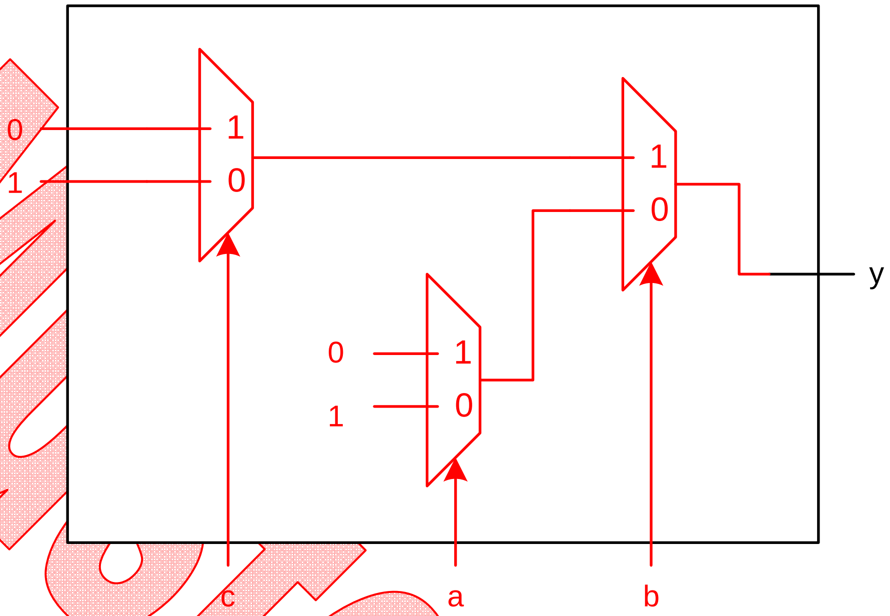
$y(0,1,1) = 1$

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

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

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

B) Die entwickelte Funktion soll mit 2:1 Multiplexern realisiert werden, wobei die Eingangsliterale a,b,c ausschließlich als Steuersignale genutzt werden sollen. Zeichnen Sie die minimale Multiplexerschaltung.



Aufgabe 2 Minimierung

Aufgabe 2.1 Primterme und KMF

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

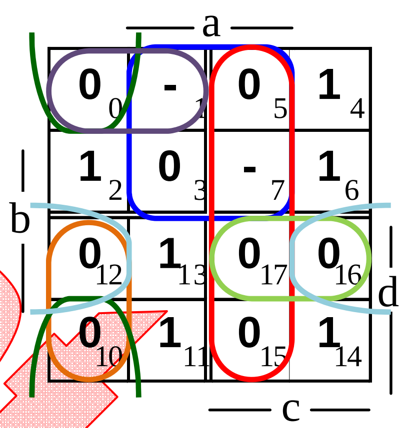


Abbildung 2-1: Symmetriediagramm

A) Geben Sie **alle** möglichen Primterme für eine vollständige Nullstellenüberdeckung der Funktion G aus Abbildung 2-1 an. Verwenden Sie zur Blockbildung auch die Freistellen.

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

B) Geben Sie eine mögliche Überdeckung mit einer minimalen Anzahl an Nullblöcken in der KMF an. Freistellen falls nötig einbezogen werden.

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

Nicht minimale Lösungen:

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

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

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

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

Aufgabe 2.2 Petrickausdruck

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

$$PA = (a \vee b) \,\&\, (a \vee c \vee e) \,\&\, (b \vee c \vee d) \,\&\, (a \vee d \vee e) \,\&\, (a \vee e \vee f) \,\&\, e$$

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 überdeckenden Größen sind durch die Präsenzvariablen a, b, c, d, e und f gegeben.
Die zu überdeckenden Größen E_i werden entsprechend der Reihenfolge wie Sie im Petrickausdruck auftauchen (links nach rechts), aufsteigend von E_1 bis E_n indiziert.

pi\Ei	E_1	E_2	E_3	E_4	E_5	E_6	
a	X	X		X	X		
b	X		X				
c		X	X				
d			X	X			
e		X		X	X	X	
f					X		

Tabelle 2-1: Überdeckungstabelle 1

B) Bestimmen Sie alle Kernspalten aus Tabelle 2-1 und markieren Sie die entsprechende(n) Zelle(n)

Kernspalte(n):

E6

Aufgabe 2.3 Verfahren nach Petrick

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

A) Wenden Sie die Spaltendominanzregel auf Tabelle 2-2 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) an.

$p_i \backslash E_i$	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
a			X					X		
b	X		X	X				X		X
c	X					X	X			X
d			X							
e		X			X	X			X	X
f	X			X	X		X			
g	X	X				X				X
h			X	X				X	X	
i	X	X			X	X				X

Tabelle 2-2: Überdeckungstabelle 2

Dominierende Spalte(n):	E3	E10	E6	E1	(E10)		
<u>Dominierte</u> Spalte(n):	E8	E2	E2	E7	(E6)		
Streichbare Spalte(n):	E3	E10	E6	E1	(E10)		

<u>Hinweis:</u> Die folgenden Teilaufgaben können und sollen vollkommen unabhängig von der vorherigen Teilaufgabe gelöst werden.

B) Wenden Sie nun die Zeilendominanz auf die bereits reduzierte Tabelle 2-3 an. Welche Zeile(n) können gestrichen werden? Streichen Sie die entsprechende(n) Zeile(n) und geben Sie die dominierte(n) und zugehörigen dominierende(n) Zeile(n), sowie die streichbaren Zeile(n) an.

$p_i \backslash N_i$	N_2	N_4	N_5	N_7	N_9	N_{11}	N_{12}	Kosten
~~a~~			~~X~~			~~X~~		~~2 GE~~
~~b~~	~~X~~		~~X~~	~~X~~			~~X~~	~~4 GE~~
~~d~~	~~X~~	~~X~~			~~X~~	~~X~~		~~6 GE~~
f			X	X		X	X	1 GE
~~g~~		~~X~~			~~X~~			~~7 GE~~
~~i~~	~~X~~			~~X~~				~~5 GE~~
~~k~~						~~X~~	~~X~~	~~3 GE~~

Tabelle 2-3: Überdeckungstabelle 3

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

C) Wenn neue Kern(e) entstanden sind geben Sie die entsprechende Kernspalt(e) an und markieren die entsprechende(n) Zelle(n) in Tabelle 2-3.

Kernspalte(n):

N4, N5, N7, N9, N12

Aufgabe 3 Optimale Codes

Ein Mikrocontroller soll für bestimmte Anwendungszwecke optimiert werden. Dabei soll auch der benötigte Speicherplatz (ROM/RAM) der Software minimiert werden, ohne dass die Funktionalität der Software eingeschränkt oder verändert werden muss.

Aus statistischen Erhebungen ergeben sich folgende Auftrittshäufigkeiten der Maschinenbefehle für die eingesetzte Software, die in der Tabelle 3-1 aufgelistet sind.

Maschinenbefehl	Auftrittshäufigkeit (%)	Ermittelte Codierung
Load	18	010
Store	16	10
Jump	22	00
Compare	17	011
Add/Sub	13	110
Mul	7	1110
Logic	4	11110
Shift	3	11111

Tabelle 3-1: Auftrittshäufigkeiten der Maschinenbefehle des Mikrocontrollers

A) Zuerst soll untersucht werden, welche mittlere Codewortlänge sich für eine Codierung ergibt, bei der alle Codewörter binär und mit <u>gleicher Länge</u> codiert würden. Geben Sie die minimale mittlere Codewortlänge unter diesen Voraussetzungen an.

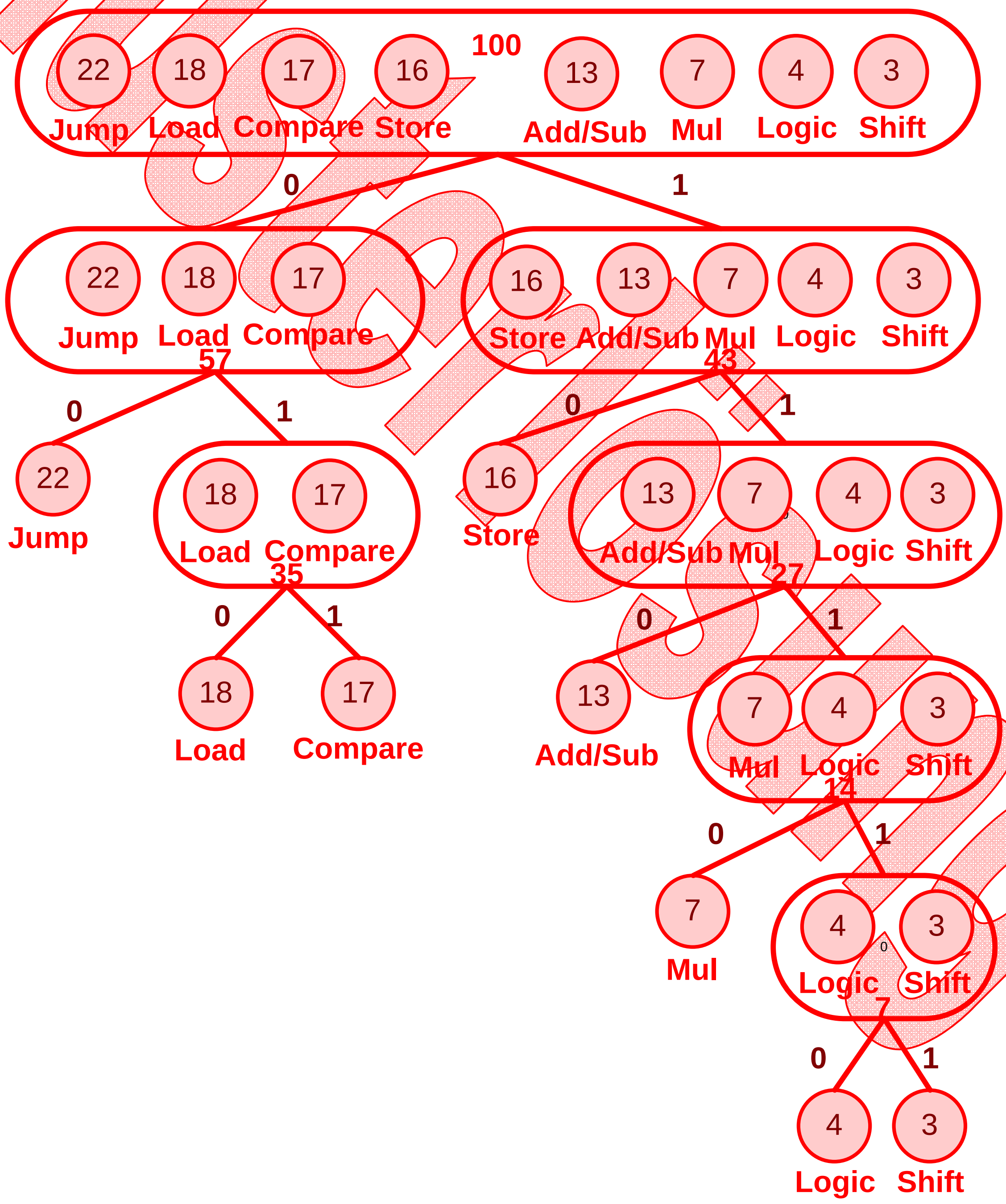
Minimale mittlere Codewörter: $\lceil ld 8 \rceil = 3$

Anhand der Auftrittshäufigkeiten aus Tabelle 3-1 soll nun eine optimale Codierung für die Maschinenbefehle entwickelt werden, die den Speicherbedarf der eingesetzten Software minimieren soll.

B) Bestimmen Sie die optimale Codierung nach dem Shannon-Fanø-Verfahren für die Auftrittshäufigkeiten aus Tabelle 3-1 und tragen Sie diese in Tabelle 3-1 ein. Der Lösungsweg muss nachvollziehbar sein!

Hinweise:

- *Sortieren Sie die Elemente zu Beginn entsprechend den Auftrittshäufigkeiten **abfallend 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“.*



C) Geben Sie die Formel zur Berechnung der mittleren Codewortlänge an. Berechnen Sie anschließend die mittlere Codewortlänge für die im Aufgabenteil B) entwickelte Codierung.

Formel: $\bar{m} = \sum_{i=1}^{n} p(x_i) * m(x_i)$

Berechnung:

$\bar{m} = 0{,}18 * 3 + 0{,}16 * 2 + 0{,}22 * 2 + 0{,}17 * 3 + 0{,}13 * 3 + 0{,}07 * 4 + 0{,}04 * 5 + 0{,}03 * 5 = 2{,}83$

D) Wie kann ermittelt werden, ob die Codierung aus Aufgabenteil B) eine ideale Lösung bezogen auf die mittlere Codewortlänge darstellt? Geben Sie eine Formel an und begründen Sie Ihre Antwort. *Berechnungen müssen nicht durchgeführt werden!*

Der minimal erreichbare Idealwert für die mittlere Codewortlänge ist der durchschnittliche Informationsgehalt: $H = \sum_{i=1}^{n} p(x_i) * ld \frac{1}{p(x_i)}$

Für ideale Lösung muss gelten: $H = \bar{m}$

Aufgabe 4 Mengen, Relationen und Graphen

Aufgabe 4.1 Fragen und Definitionen

A) Seien A und B endliche Mengen und A echte Teilmenge/Untermenge von B. Welche Aussage lässt sich daraus über die relative Mächtigkeit der Mengen ableiten? Wie unterscheidet sich die „echte Untermenge“ von der allgemeinen „Untermenge“?

Die Mächtigkeit von A ist kleiner der von B.

Bei einer allgemeinen Untermenge B muss die Mächtigkeit nicht unbedingt kleiner sein als die von A. Bei einer echten Untermenge hingegen schon.

Mächtigkeit UM: $|A| \geq |B|$; Mächtigkeit echte UM: $|A| > |B|$

B) Was ist der Unterschied zwischen einer einfachen, offenen Kantenprogression und einer Wegprogression auf einem gerichteten Graphen?

In einer Wegprogression kommt jede Kante höchstens ein Mal vor, während bei der einfachen, offenen Kantenprogression jeder Knoten nur ein Mal enthalten sein kann. Jede einfache, offene Kantenprogression ist damit eine Wegprogression, nicht jedoch umgekehrt!

C) Welche Eigenschaften muss eine Verträglichkeitsrelation erfüllen?

Reflexiv, Symmetrisch, nicht transitiv.

Aufgabe 4.2 Graphen

Gegeben sei folgender Graph als Grundlage einer Planungsstrategie. Gesucht ist die maximale Durchführungszeit eines Prozesses (worst case execution time).

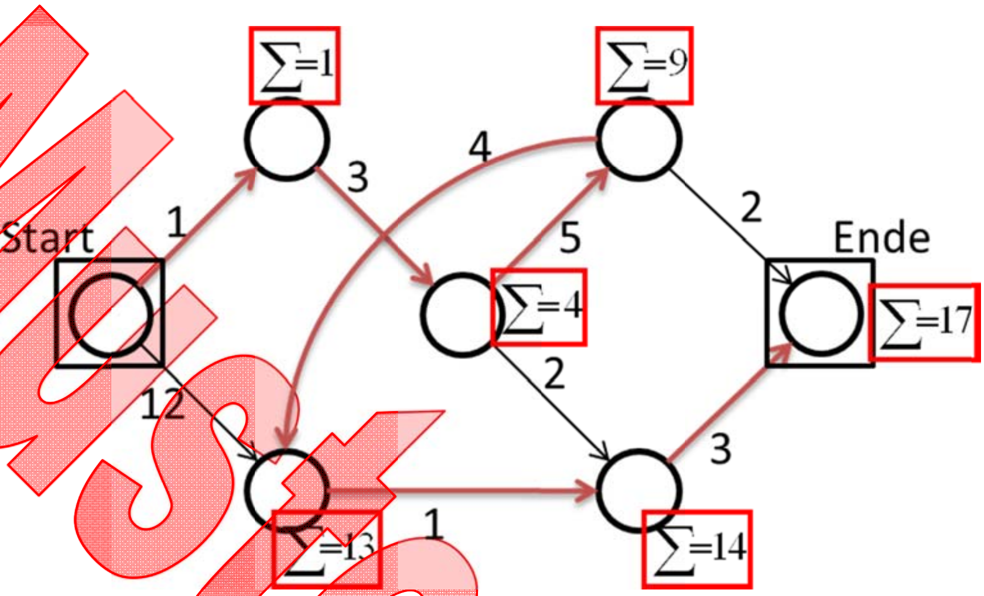


Abbildung 4-1: Gerichteter und gewichteter Graph

A) Führen Sie die in der Vorlesung eingeführte Planungsstrategie (Längster-Weg-Suche) auf dem Graphen aus Abbildung 4-1 durch. Dazu sind Start- und Endknoten schon markiert. Markieren Sie die nötigen Kanten im Graphen. Geben Sie die **maximale** Durchführungszeit auch für die Teilschritte an.

Maximale Durchführungszeit: 17

B) Welche Eigenschaft muss ein Graph erfüllen, damit ein dualer Graph dazu konstruiert werden kann?

Planarität

Gegeben sei nun folgender Graph:

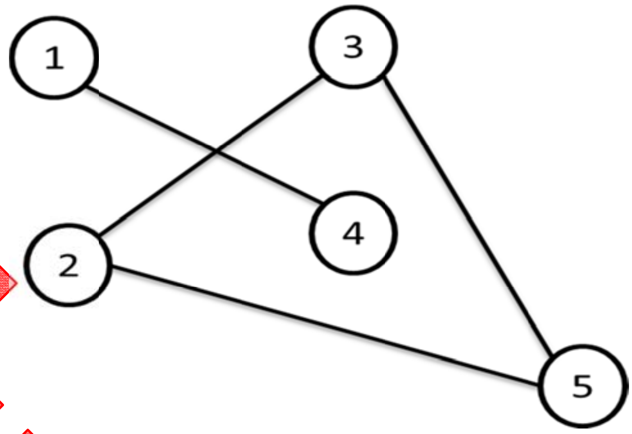
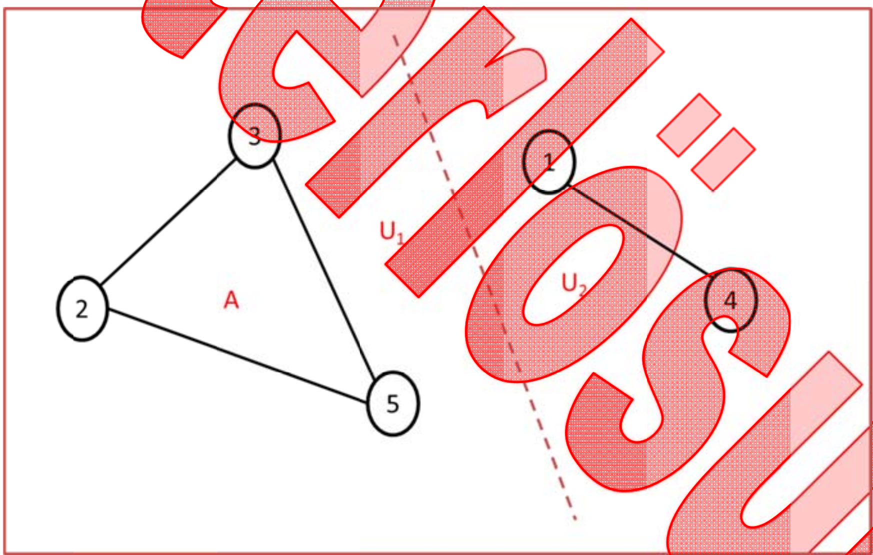


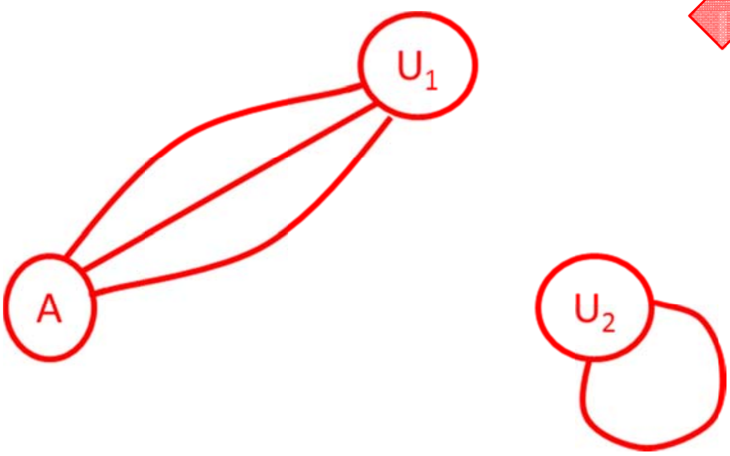
Abbildung 4-2: Ausgangsgraph

C) Konstruieren Sie zum gegebenen Graphen aus Abbildung 4-2 einen dualen Graphen. Wählen Sie dabei ggf. zunächst eine geeignete Darstellung des Graphen. Benennen/kennzeichnen Sie die entsprechenden Gebiete und geben Sie den dualen Graphen an.

Zunächst kreuzungsfreie (planare) Darstellung:



Dann dazu dualer Graph:



Aufgabe 5 Polyadische Zahlensysteme

Aufgabe 5.1 BCD

A) Addieren Sie die im Dezimalsystem gegebenen Zahlen 8657_D und 4943_D im BCD Code. Stellen Sie ihren Lösungsweg – inklusive aller notwendigen Korrekturschritte – ausführlich dar.

		BCD				Dezimalsystem
		1000	0110	0101	0111	8657
+		0100	1001	0100	0011	+4943
		----	----	1---	111-	
=		1100	1111	1001	1010	
Korr. wegen:		Ps.	Ps.		Ps.	
		1100	1111	1001	1010	
+		0110	0110		0110	
		1--1	11--	--11	11--	
=	0001	0011	0101	1010	0000	
Korr. wegen:				Ps.		
		0011	0101	1010	0000	
+				0110		
		----	--11	11--	----	
=	0001	0011	0110	0000	0000	13600

Aufgabe 5.2 Konvertierung

A) Wandeln Sie die gegebenen Zahlen aus Tabelle 5-1 in das angegebene Zahlensystem um. Geben Sie Ihre Rechenschritte eindeutig an.

Gegebene Zahl	Ziel-Zahlensystem	Rechenweg	Konvertierte Zahl
58_D	5	58_D : 5 = 11 Rest 3 11_D : 5 = 2 Rest 1 2_D : 5 = 0 Rest 2	213_5
132_O	9	Dezimal: 132_O = 1*64 + 3*8 + 2 = 90_D 90_D : 9 = 10 Rest 0 10_D : 9 = 1 Rest 1 1_D : 9 = 0 Rest 1	110_9

Tabelle 5-1: Konvertierungstabelle 1

B) Vervollständigen Sie die Tabelle 5-2, indem Sie die offenen Felder durch die entsprechend konvertierten Zahlen ergänzen.

Dezimal	Binär	Oktal	BCD
329_D	1 0100 1001_B	511_O	0011 0010 1001_{BCD}
864_D	11 0110 0000_B	1540_O	1000 0110 0100_{BCD}

Tabelle 5-2: Konvertierungstabelle 2

Aufgabe 6 Automaten

Aufgabe 6.1 Erstellen eines endlichen Automaten

Es soll die Steuerung eines Getränkeautomaten entworfen werden. Der Automat akzeptiert 50 ct- und 1 €-Münzen und gibt ein Getränk aus, sobald der Getränkepreis von 1,50 € eingeworfen und anschließend die Anforderungstaste gedrückt wurde. Er besitzt ein Display, das den bereits eingeworfenen Betrag anzeigt. Die Anzeige des Betrags wird auf 0,00 € zurückgesetzt, sobald das Getränk ausgegeben wird.

Übersteigt der eingeworfene Betrag nach Einwurf einer Münze den Getränkepreis, so wird die Münze zurückgegeben. Auf dem Display erscheint dann die Meldung „ungültiger Betrag". Drückt man die Anforderungstaste, bevor der vollständige Getränkepreis eingeworfen wurde, erscheint auf dem Display „Geld nachwerfen".

Die Automatensteuerung besitzt folgende Ports:

Eingänge	
Ew1	Einwurf einer 50ct-Münze
Ew2	Einwurf einer 1€-Münze
Anf	Anfordern des Getränks
Ausgänge	
SD	Status-Display, *siehe Tabelle auf Seite 21*
RG	Rückgabe der zuletzt eingeworfenen Münze
AG	Ausgabe des Getränks

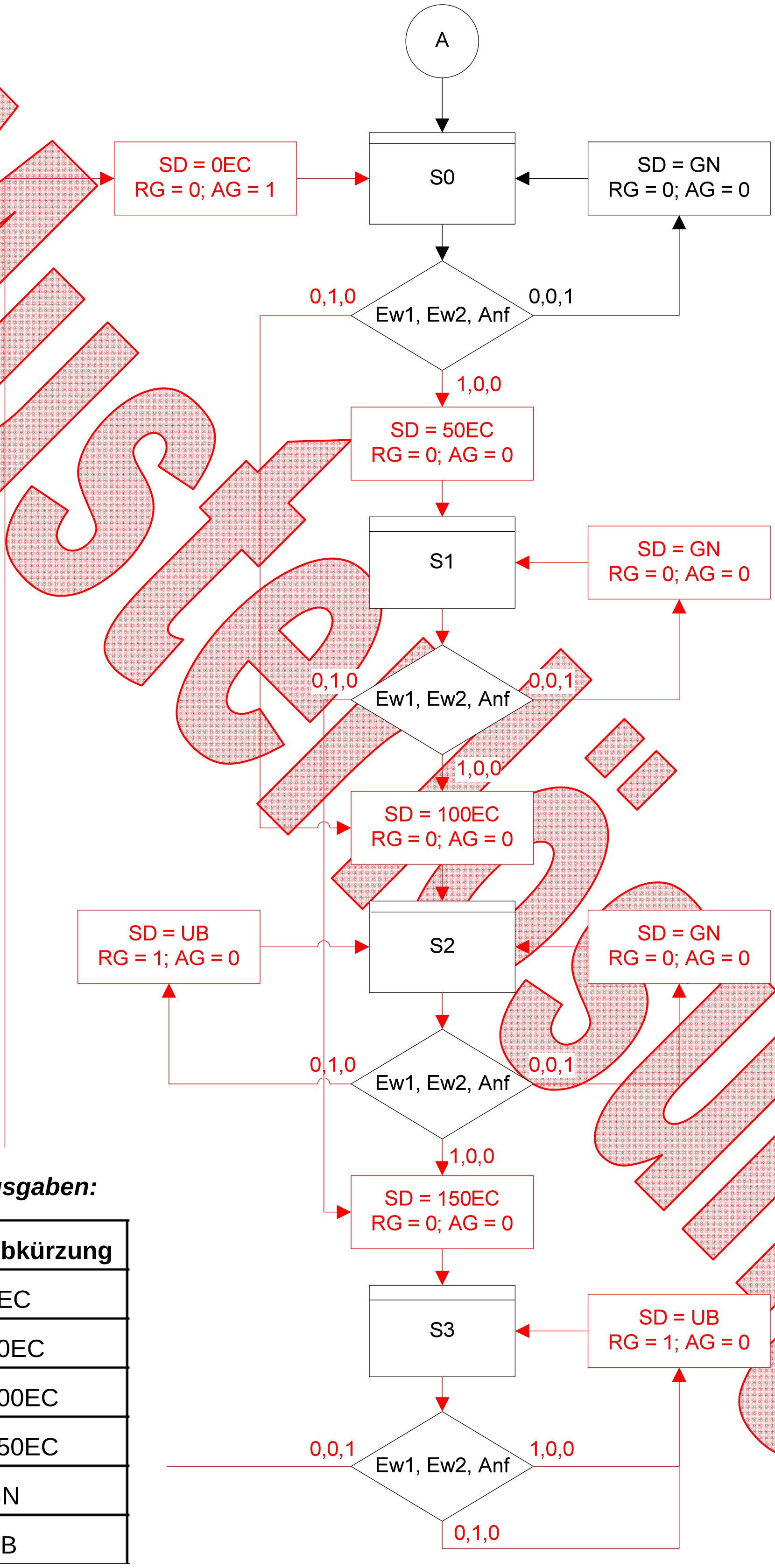
A) Realisieren Sie das Ablaufdiagramm als Mealy-Automat. Verwenden Sie hierbei nur die oben beschriebenen Signale und nehmen Sie an, dass die Signale *Ew1*, *Ew2* und *Anf* nicht gleichzeitig aktiv sein können. Stellen Sie die Ausgaben als Blöcke dar. Verwenden Sie die vorgegebene Abbildung 6-1 und die gegebenen Abkürzungen für die Display-Ausgaben (siehe folgende Seite).

B) Wie viele JK-Flip-Flops werden mindestens benötigt, um den Automaten aus Aufgabenteil A) zu implementieren?

4 Zustände → ld 4 = 2 Flip-Flops

C) Ist der Automat auch als Moore-Automat realisierbar? Begründen Sie Ihre Antwort.

Ja, jeder Mealy-Automat kann in einen äquivalenten Moore-Automaten überführt werden, wenn entsprechend viele Zwischenzustände eingefügt werden.



Mögliche Display-Ausgaben:

Ausgabe	**Abkürzung**
0,00 €	0EC
0,50 €	50EC
1,00 €	100EC
1,50 €	150EC
Geld nachwerfen	GN
Ungültiger Betrag	UB

Abbildung 6-1: Ausgangsgraph des Automaten

Aufgabe 6.2 Analyse eines Automaten

Gegeben sei die Zustandsübergangstabelle eines Schaltwerks mit drei Speicherelementen.

Die Zustände des Schaltwerks werden direkt zur weiteren Nutzung ausgegeben. Es werden zwei vorderflanken-gesteuerte JK-Flip-Flops und ein T-Flip-Flop zur Speicherung der Zustände verwendet. FF_0 enthält hierbei den Wert von Q_0, FF_1 den Wert von Q_1 und FF_2 den Wert von Q_2.

Die Zustandsübergänge sind in der folgenden Tabelle gegeben:

	Q^{υ}			$Q^{\upsilon+1}$			FF_2		FF_1		FF_0	Ausgabe		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	T_0	y_2	y_1	y_0
0	0	0	0	0	0	1	0	-	0	-	1	0	0	0
1	0	0	1	0	1	1	0	-	1	-	0	0	0	1
2	0	1	0	1	1	0	1	-	-	0	0	0	1	0
3	0	1	1	0	1	0	0	-	-	0	1	0	1	1
4	1	0	0	0	0	0	-	1	0	-	0	1	0	0
5	1	0	1	1	0	0	-	0	0	-	1	1	0	1
6	1	1	0	1	1	1	-	0	-	0	1	1	1	0
7	1	1	1	1	0	1	-	0	-	1	0	1	1	1

Tabelle 6-1: Zustandstabelle

A) Vervollständigen Sie in Tabelle 6-1 die Ansteuerfunktionen für FF_0-FF_2. Realisieren Sie die Ansteuerung jedes JK-Flip-Flops in jedem Zustand für die spätere Minimierung mit mindestens einer Don't-Care-Stelle.

B) Welcher spezielle Zähler wird durch die Zustandstabelle (Tabelle 6-1) beschrieben?

Zyklischer 3-bit-Greycode-Zähler

C) Um welchen Automatentyp handelt es sich? Begründen Sie Ihre Antwort.

Es handelt sich um einen Medwedew-Automaten.

Begründung: Ausgabe = Zustand

D) Bestimmen und minimieren Sie die Ansteuerfunktionen von FF_0 und FF_2. Nutzen Sie hierbei die Don't-Care-Stellen in der Ansteuerung der JK-Flip-Flops aus. Geben Sie die entsprechenden algebraischen Ausdrücke an.

J_2

		Q_0	Q_0	
	0	0	-	-
Q_1	1	0	-	-
			Q_2	Q_2

$$J_2 = \overline{Q_0}\, Q_1$$

K_2

		Q_0	Q_0	
	-	-	0	1
Q_1	-	-	0	0
			Q_2	Q_2

$$K_2 = \overline{Q_0}\;\overline{Q_1}$$

T_0

		Q_0	Q_0	
	1	0	1	0
Q_1	0	1	0	1
			Q_2	Q_2

$$T_0 = \overline{Q_0}\;\overline{Q_1}\;\overline{Q_2} + Q_0 Q_1 \overline{Q_2} + Q_0 \overline{Q_1} Q_2 + \overline{Q_0} Q_1 Q_2$$

Tabelle 6-2: KV-Diagramme

Aufgabe 7 CMOS-Schaltnetze

Bei einem CMOS-Schaltkreis ist nur das Pull-Up-Schaltnetz vorhanden. In mehreren Teilaufgaben soll nun die Pull-Down-Funktion bestimmt und das entsprechende Schaltnetz realisiert werden.

Gegeben sei folgender CMOS-Schaltnetz:

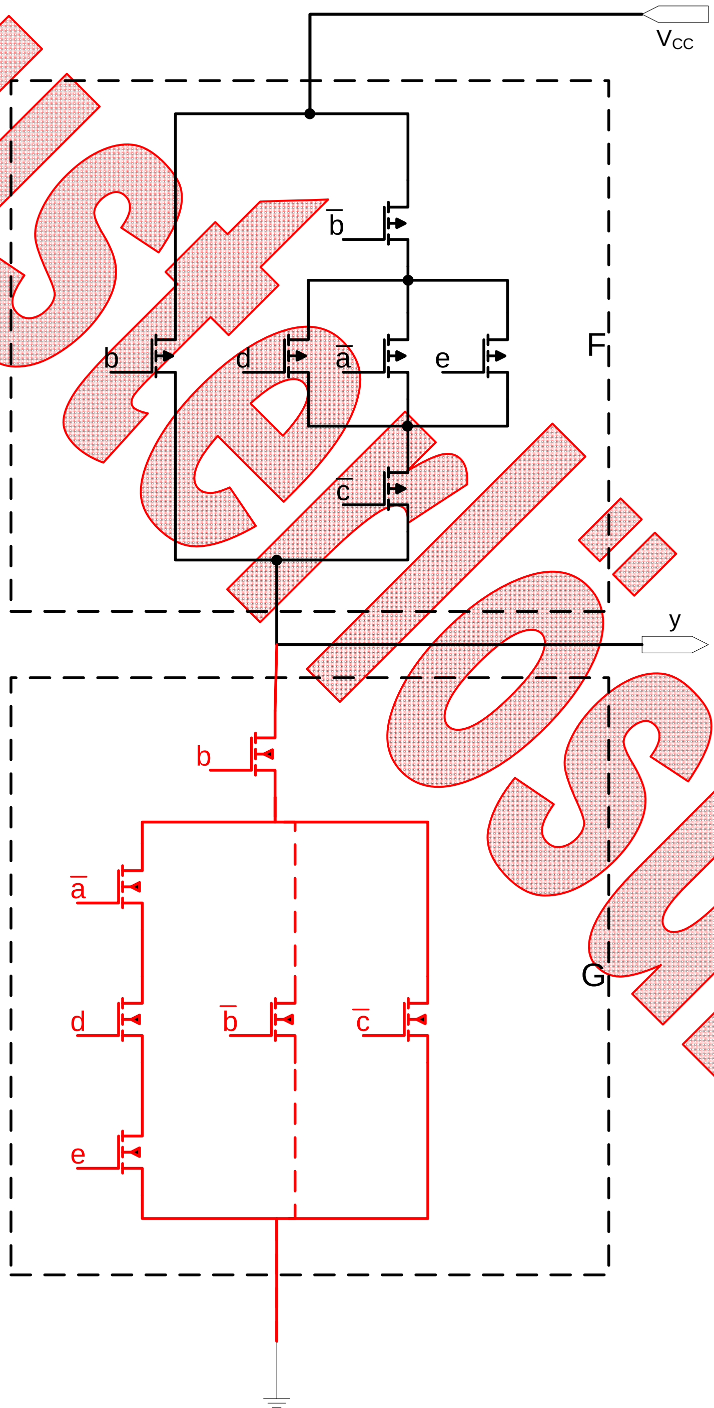


Abbildung 7-1: CMOS-Schaltnetze

A) Geben Sie für die vorliegende Schaltung die Pull-Up-Funktion F in disjunktiver Minimalform (DMF) an.

$$F = \overline{b} + b\left(a + \overline{d} + \overline{e}\right)c = \overline{b} + abc + bc\overline{d} + bc\overline{e}$$

B) Welche Bedingungen müssen die Pull-Up- (F) / Pull-Down-Funktionen (G) erfüllen, damit das CMOS-Schaltnetz für alle möglichen Eingangsbelegungen einen definierten Ausgangspegel hat?

Kurzschlussfreiheit: (F & G) = 0

Wohldefiniertheit: (F + G) = 1

C) Gegeben ist nun eine Pull-Up-Funktion F'. Bestimmen Sie für diese Pull-Up-Funktion die dazugehörende Pull-Down-Funktion G', so dass die CMOS-Schaltung wohldefiniert und kurzschlussfrei ist. Geben Sie G' in der DMF an.

$$F' = c(a\overline{b}e + cd)$$

$$G' = \overline{F'} = \overline{c(a\overline{b}e + cd)}$$

$$= \overline{c} + \overline{a\overline{b}e + cd}$$

$$= \overline{c} + (\overline{a\overline{b}e} \times \overline{cd})$$

$$= \overline{c} + (\overline{a} + b + \overline{e}) \times (\overline{c} + \overline{d})$$

$$= \overline{c} + \cancel{\overline{a}}\overline{c} + \cancel{b}\overline{c} + \cancel{\overline{c}}\overline{e} + \overline{a}\,\overline{d} + b\overline{d} + \overline{d}\,\overline{e}$$

D) Gegeben ist nun die Pull-Down-Funktion G. Vervollständigen sie das CMOS-Schaubild in Abbildung 7-1, in dem Sie die Pull-Down-Funktion unter Verwendung von nMOS-Feldeffekttransistoren einzeichnen.

$$G = b(\overline{b} + \overline{c} + \overline{a}de)$$

Aufgabe 8 Schaltnetze

Aufgabe 8.1 Strukturausdruck und PAL-Realisierung

Gegeben ist das Schaltnetz in Abbildung 8-1.

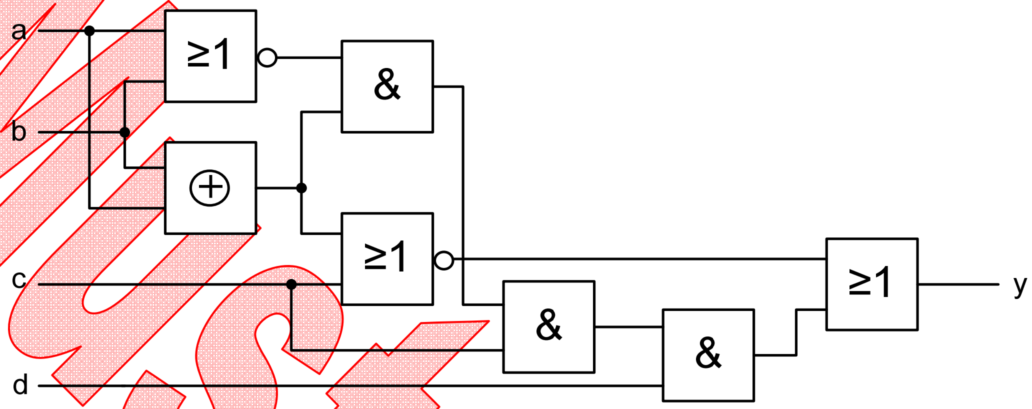


Abbildung 8-1: Schaltnetz mit Grundgattern und Komplexgattern

A) Ermitteln Sie den Strukturausdruck y des in Abbildung 8-1 gegebenen Schaltnetzes.

Hinweis: Der Ausdruck muss nicht vereinfacht werden. Sowohl einfache, als auch komplexe boolesche Operatoren dürfen verwendet werden.

$$y = \left[\overline{(a+b)} \,\&\, (\mathrm{a} \oplus \mathrm{b}) \,\&\, c \,\&\, d\right] + \left[\overline{(\mathrm{a} \oplus \mathrm{b}) + c}\right]$$

B) Wandeln Sie den in Teilaufgabe A) gefundenen Strukturausdruck so um, dass ein minimales zweistufiges UND/ODER-Schaltnetz entsteht, das sich anschließend als PAL realisieren lassen würde. Der Lösungsweg muss nachvollziehbar sein.

$$y = \left[\overline{(a+b)} \,\&\, (a \oplus b) \,\&\, c \,\&\, d\right] + \left[\overline{(a \oplus b) + c}\right]$$

XOR-Verknüpfung ersetzen:

$$y = \left[\overline{(a+b)} \,\&\, \left((a \,\&\, \overline{b}) + (\overline{a} \,\&\, b)\right) \,\&\, c \,\&\, d\right] + \left[\overline{(a \,\&\, \overline{b}) + (\overline{a} \,\&\, b) + c}\right]$$

Umformung nach DeMorgan:

$$y = \left[(\overline{a} \,\&\, \overline{b}) \,\&\, \left((a \,\&\, \overline{b}) + (\overline{a} \,\&\, b)\right) \,\&\, c \,\&\, d\right] + \left[\overline{(a \,\&\, \overline{b})} \,\&\, \overline{(\overline{a} \,\&\, b)} \,\&\, \overline{c}\right]$$

Kürzen wegen $\left[(\overline{a} \,\&\, \overline{b}) \,\&\, \left((a \,\&\, \overline{b}) + (\overline{a} \,\&\, b)\right) \,\&\, c \,\&\, d\right] = 0$

$$y = \left[\overline{(a \,\&\, \overline{b})} \,\&\, \overline{(\overline{a} \,\&\, b)} \,\&\, \overline{c}\right]$$

Nochmal nach DeMorgan umformen:

$$y = \left[(\overline{a} + b) \,\&\, (a + \overline{b}) \,\&\, \overline{c}\right]$$

UND/ODER Schaltnetz für PAL-Realisierung:

$$y = (\overline{a} \,\&\, \overline{b} \,\&\, \overline{c}) + (a \,\&\, b \,\&\, \overline{c})$$

C) Geben Sie nun die PAL-Realisierung des gefundenen minimalen Ausdrucks aus Teilaufgabe B) an. Verwenden Sie dazu Abbildung 8–2.

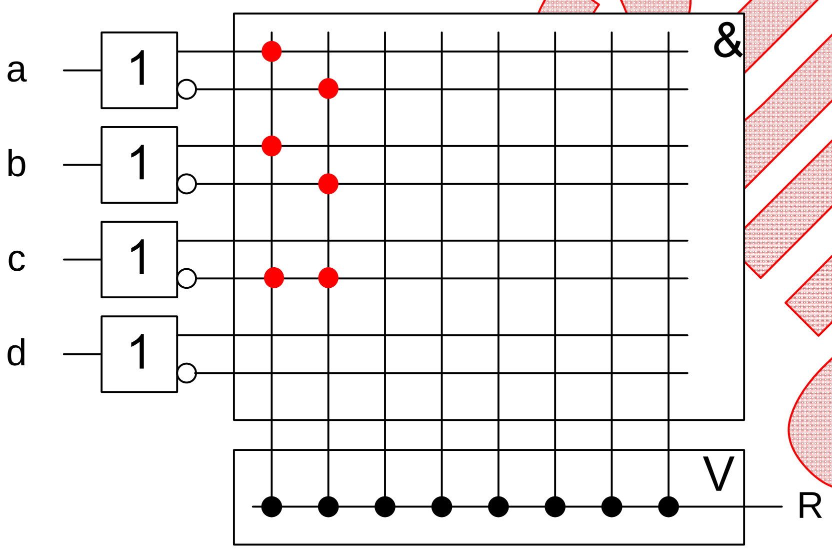


Abbildung 8–2: PAL-Schaltnetz

Aufgabe 8.2 3:1 Multiplexer-Realisierung

Sie sollen nun in mehreren Schritten einen 3:1 Multiplexer aus NAND-Gattern (Full-NAND) realisieren. Das Blockschaltbild und die Schaltfunktion des Multiplexers sind in Abbildung 8–3 dargestellt. Die Eingangssignale a, b und c haben eine Breite von einem Bit.

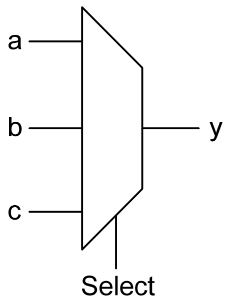


Abbildung 8–3: Blockschaltbild des 3:1 Multiplexer

A) Welche Bitbreite ist minimal für die Ansteuerung des Select-Signals des zu entwerfenden 3:1 Multiplexer nötig?

Minimal nötige Bitbreite des Select-Signals $\lceil ld(3) \rceil = 2$

B) Geben Sie die Schaltfunktion y des Multiplexers in Abhängigkeit der Eingänge a, b, c und Select. Verwenden Sie die in der vorherigen Teilaufgabe ermittelte Bitbreite für das Select-Signal. Jeder Eingang soll dabei mit genau einer Kombination der Select-Bits durchgeschaltet werden, für alle übrigen Select-Bitkombinationen soll am Ausgang y eine „0“ anliegen. Verwenden Sie folgende Bezeichnung für die Bits des Select-Eingangs: s0, s1, s2 …

$$y = \left(a \,\&\, \overline{s0} \,\&\, \overline{s1}\right) + \left(b \,\&\, s0 \,\&\, \overline{s1}\right) + \left(c \,\&\, \overline{s0} \,\&\, s1\right)$$

C) Formen Sie die gefundene Schaltfunktion so um, dass die Schaltung ausschließlich mit NAND3-Gattern (NAND-Gatter mit drei Eingängen) realisiert werden kann.

$$y = \left(a \,\&\, \overline{s0} \,\&\, \overline{s1}\right) + \left(b \,\&\, s0 \,\&\, \overline{s1}\right) + \left(c \,\&\, \overline{s0} \,\&\, s1\right)$$

Doppelte Negation:

$$y = \overline{\overline{\left(a \,\&\, \overline{s0} \,\&\, \overline{s1}\right) + \left(b \,\&\, s0 \,\&\, \overline{s1}\right) + \left(c \,\&\, \overline{s0} \,\&\, s1\right)}}$$

Umformung nach DeMorgan:

$$y = \overline{\overline{\left(a \,\&\, \overline{s0} \,\&\, \overline{s1}\right)} \,\&\, \overline{\left(b \,\&\, s0 \,\&\, \overline{s1}\right)} \,\&\, \overline{\left(c \,\&\, \overline{s0} \,\&\, s1\right)}}$$

Umformung negierter Eingangsliterale in NAND3-Gatter:

$$y = \overline{\overline{\left(a \,\&\, \overline{(s0 \,\&\, s0 \,\&\, s0)} \,\&\, \overline{(s1 \,\&\, s1 \,\&\, s1)}\right)} \,\&\, \overline{\left(b \,\&\, s0 \,\&\, \overline{(s1 \,\&\, s1 \,\&\, s1)}\right)} \,\&\, \overline{\left(c \,\&\, \overline{(s0 \,\&\, s0 \,\&\, s0)} \,\&\, s1\right)}}$$

D) Zeichnen Sie die Schaltung des 3:1 Mulitplexers in Full-NAND-Technik. Verwenden Sie dazu eine minimale Anzahl an NAND3-Gattern.