Klausur (SS 2018)
Digitaltechnik

Institut für Technik der Informationsverarbeitung
Prof. Dr.-Ing. Dr. h. c. Jürgen Becker

Klausur: Digitaltechnik
Datum: 30. August 2018

Teilnehmer:
Matr.-Nr.:
ID:

Hörsaal:
Platz:

Es gelten die folgenden Regelungen:

- Die Bearbeitungszeit beträgt 120 Minuten.
- Es sind keine Hilfsmittel erlaubt, außer
  - der beigelegten dreiseitigen Formelsammlung und
  - einer selbst geschriebene zweiseitigen Formelsammlung im DIN A4 Format.
- Nutzen Sie nur **dokumentenechte Schreibgeräte** – keine Bleistifte oder rote Farbe!
- Die Verwendung von eigenem Papier ist nicht zugelassen.
- Vermeiden Sie das Beschreiben der Rückseiten.
- Bei Bedarf erhalten Sie Zusatzblätter von der Aufsicht.
  - Versehen Sie solche Blätter unbedingt mit Ihrer Matrikelnummer.
  - Ordnen Sie jedes zusätzliche Lösungsblatt einer Aufgabe eindeutig zu.

Die vorliegende Klausur besteht aus **34 Blättern** und einer dreiseitigen Formelsammlung.

		Seite	≈ Pkt. in %	Punkte
Aufgabe 1:	Allgemeine Fragen	2	12	**15**
Aufgabe 2:	Zahlensysteme	5	13	**16**
Aufgabe 3:	Boolsche Algebra	9	11	**14**
Aufgabe 4:	Minimierung	13	13	**16**
Aufgabe 5:	Mengen, Relationen und Graphen	16	13	**16**
Aufgabe 6:	Optimale Codes	19	10	**13**
Aufgabe 7:	Automaten	23	14	**18**
Aufgabe 8:	CMOS und Gatter	28	12	**15**
				$\sum$ **123**

Aufgabe 1: Allgemeine Fragen

15

Aufgabe 1.1: Information und Codierung

A) Eine Grafikkarte soll die drei Farbanteile eines Pixels (Rot, Grün, Blau) durch jeweils einen Wert im Bereich von 0...220 angeben. Wie groß muss der Speicher der Grafikkarte mindestens sein, damit ein Bildschirm mit der Auflösung von 800x600 Pixeln angesteuert werden kann? Das Endergebnis muss nicht ausgerechnet werden. **2**

Codierung eines Wertebereichs [0;220] -> ld(221) = 8
–> 8Bit * 3Farben = 24Bit pro Pixel
800*600*24Bit (=480.000*24Bit = 11.520.000Bit = 1,44MByte)

B) Der Speicher sei nun 30 kByte groß. Welche maximale Bildschirmauflösung lässt sich mit der Grafikkarte ansteuern, falls das Seitenverhältnis 1:1 betragen soll? **1**

da 8 Bit pro Farbe –> 3Byte pro Pixel –> 30MByte/3Byte = 10.000 Pixel
Seitenverhältnis 1:1 –> $\sqrt{10.000} = 100$
–> Eine Auflösung von 100x100 Pixeln kann maximal realisiert werden.

C) Wie viele gültige Codewörter gibt es bei einem 2-aus-4 Code? Geben Sie explizit alle gültigen Codewörter an. **1**

es gibt 6 gCW
–> 0011; 1100; 0110; 1001; 1010; 0101

D) Wie viele gültige Codewörter kann es bei einem 2-aus-4 Code maximal geben, wenn ein 1-Bit-Fehler korrigiert werden soll? Begründen Sie Ihre Antwort. **1**

Um 1-Bit-Fehler korrigieren zu können, wird eine HDmin von 3 benötigt
aus den 6 gCWs besitzen immer nur zwei CWs eine HDmin von >= 3
–> Es gibt nur 2 gCWs. (z.B.: 0011 und 1100)

E) Ist bezüglich der Fehlererkennung bzw. Fehlerkorrektur ein 6-aus-9 Code oder ein 3-aus-10 Code zu bevorzugen? Begründen Sie Ihre Antwort. **1**

Es macht kein Unterschied, da mit beiden nur 1-Bit-Fehler erkannt werden kann.

Aufgabe 1.2: Zahlensysteme

In einem Paralleluniversum haben die Menschen zwei Hände mit jeweils 16 Fingern, weswegen sie im Duotrigesimalzahlensystem (**Basis 32**) rechnen. Die Codierung der Dezimalzahlen ins Duotrigesimalzahlensystem ist in Tabelle 1.1 gegeben.

Dezimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Duotrigesimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dezimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Duotrigesimal	G	H	J	K	M	N	P	Q	R	S	T	V	W	X	Y	Z

Tabelle 1.1: Symbole im Duotrigesimalzahlensystem nach Crockford

A) Formen Sie die Duotrigesimal-Zahl (**Basis 32**) Z in das Binärsystem (**Basis 2**) um. **1**

Z entspricht laut Tabelle 31 in Dezimal und somit in Binär dargestellt 11111

B) Formen Sie die vier-stellige Duotrigesimal-Zahl (**Basis 32**) S18Y in das Binärsystem (**Basis 2**) um. **1**

S entspricht laut Tabelle 25 in Dezimal und somit in dargestellt in Binär 11001
–> die komplette Duotrigesimal-Zahl in Binär: 11001 00001 01000 11110

C) Formen Sie nun die die 20-stellige Binärzahl (**Basis 2**) 11101 01000 10001 10100 in das Duotrigesimalzahlensystem (**Basis 32**) um. **2**

11101 entspricht 29 in Dez und damit X in Duotrigesimal
–> die Duotrigesimal-Zahl lautet also: X8HM

D) Auch die Menschen im Paralleluniversum können multiplizieren. Wie sieht die Multiplikation der Duotrigesimalzahl A82 mit der Duotrigesimalzahl 23 aus? Rechnen Sie nachvollziehbar im Duotrigesimalzahlensystem.

2

	A	8	2		* 23 =
	M	G	4		+
		Y	R	6	+
Übertrag	1				
	N	E	W	6	

E) Ebenfalls kennen die Menschen im Paralleluniversum auch die Exponentialdarstellung. Schreiben Sie die Duotrigesimalzahl $5,H * 10^A$ in ganzahliger Darstellung zur Basis 32.

1

Verschiebung um A = 10 Stellen
–> 5 H0000 00000

F) Wie viele Duotrigesimal-Symbole werden benötigt um eine 125-stellige Hexadezimalzahl im Duotrigesimalzahlensystem darzustellen?

2

125 * 4 = 500 Bits –> 500 / 5 = 100 Duotrigesimal-Symbole
oder kurz: 125 * 4/5 = 100 Duotrigesimal-Symbole

Aufgabe 2: Zahlensysteme

16

Aufgabe 2.1: Umrechnung von Zahlensystemen

A) Vervollständigen Sie die Tabelle 2.1, indem Sie die offenen Felder durch die entsprechende Konvertierung ergänzen.

6

Hexadezimal	Dezimal	Oktal	Binär
BE_H	190_D	276_O	$1001\ 1110_B$
$29A_H$	666_D	1232_O	$0010\ 1001\ 1010_B$
$1B6_H$	$438D$	666_O	$0001\ 1011\ 0110_B$
$CA3_H$	3235_D	6243_O	$1100\ 1010\ 0011_B$

Tabelle 2.1: Umrechnung von Zahlensystemen

Aufgabe 2.2: Fließkommazahl

Abbildung 2.1 zeigt eine Darstellung von Fließkommazahlen nach dem IEEE-754-2008-Standard mit halber Genauigkeit (16 Bit). Das höchstwertige Bit stellt das Vorzeichen V dar, die nachfolgenden fünf Bits den Exponenten E und die niederwertigsten zehn Bits die Mantisse M.

V	E4	E3	E2	E1	E0	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0

Abbildung 2.1: 16-Bit-Fließkommazahlenformat

A) Für die Berechnung von mathematischen Operationen legt ein System die Daten folgendermaßen im Speicher ab: 5

Operator (8 Bit)	**Literal I (16 Bit)**	**Literal II (16 Bit)**

Die Daten werden wie folgt miteinander verknüpft:

[Literal I] [Operator (ASCII)] [Literal II]

1. Berechnen Sie die beschriebene Operation unter Verwendung des **Binärsystem**
2. Bestimmen Sie den Wert der Fließkommazahlen im **Dezimalsystem** und berechnen Sie die Operation

Verwenden Sie für die Berechnungen folgenden Speicherinhalt:

2B49004200$_H$

Hinweis: Verwenden Sie 16-Bit-Fließkommazahlen nach Abb. 2.1. Der Operator ist ASCII codiert.

$2B_H \mathrel{\hat=} {}'+'$
$4900_H \mathrel{\hat=} 0100\ 1001\ 0000\ 0000_B \mathrel{\hat=} +1.01_B * 2^{18-15} = 1,25 * 2^3 = 10$
$4200_H \mathrel{\hat=} 0100\ 0010\ 0000\ 0000_B \mathrel{\hat=} +1.10_B * 2^{16-15} = 1,50 * 2^1 = 3$

$1.01_B * 2^3 => 1.0100\ 0000\ 00\ 1.10_B * 2^1 => 0.0110\ 0000\ 00(rechtsschiebenum 3-1 = 2)$
101 0000 0000
+ 001 1000 0000
− − − − − − − −−
110 1000 0000
Kein Übertag => Exponent unverändert

=> $0100\ 1010\ 1000\ 0000_B$
10 + 3 = 13

B) Geben Sie die Zahl 3,875 als binäre Festkommazahl an, welche 8 Vorkomma- und 8 Nachkommastellen besitzt. **1**

$0000\ 0011\ 1110\ 0000_B$

Aufgabe 2.3: BCD-Codes

A) Geben Sie an, wieviele Korrekturschritte der Stibitz-Code bei der Addition mindestens und maximal benötigt. **1**

Bei der Addition wird immer genau ein Korrekturschritt benötigt.

B) Berechnen Sie mit Hilfe des BCD-Codes folgende Aufgabe: **3**

7 - 42

07 => 0000 0111
42 => 0100 0010
Komplement 100 - 42 = 58 => 0101 1000
0000 0111
0101 1000 +
———
0101 1111 Korrektur
0000 0110 +
———-
0110 0101 => kein Übertrag Ergebnis ist negativ
———
0110 0101 = 65
Komplement
100 - 65 => 35
Ergebniss = -35

14

Aufgabe 3: Boolsche Algebra

Aufgabe 3.1: Axiome und Regeln der Schaltalgebra

A) Was unterscheidet die Axiome von den Regeln der Schaltalgebra?

1

Ein Axiom ist eine grundlegende Aussage, die ohne Beweis angenommen wird. Alle anderen Regeln lassen sich dagegen aus den Axiomen ableiten.

B) Gegeben sei die Boolsche Algebra $BA = [K, \top, \bot, \bar{\ }, O, I]$ nach Huntington. Wie lässt sich die Schaltalgebra SA auf die gegebene BA abbilden? Vervollständigen Sie die untenstehende Tabelle.

1

BA:	K	$\top$	$\bot$	$\bar{\ }$	O	I
SA:	$\{0, 1\}$	$\wedge$	$\vee$	$\bar{\ }$	0	1

C) Was sagt das Dualitätsprinzip der Boolschen Algebra $BA = [K, \top, \bot, \bar{\ }, O, I]$ aus?

1

Alles Sätze der Boolschen Algebra bleiben gültig, wenn man $\top$ mit $\bot$ sowie O mit I vertauscht.

D) Zeigen Sie formal, dass das Absorptionsgesetz $(a \wedge b) \vee a = a$ der Schaltalgebra gültig ist. Führen Sie dazu die linke Seite der Gleichung auf a zurück. Tragen Sie die Einzelschritte zeilenweise in die nachfolgende Tabelle ein und geben Sie jeweils die Nummer des verwendeten Satzes gemäß dem Formelblatt an.
(*Hinweise:* Die Verwendung des Kommutativgesetzes (H2) muss nicht explizit als Einzelschritt angegeben werden. Von allen anderen Axiomen/Regeln darf nur eine pro Einzelschritt angewendet werden. Die Regeln *R11a* und *R11b* dürfen für diesen Beweis naturgemäß nicht verwendet werden.)

4

Regel	Term
-	$(a \wedge b) \vee a$
R6b / H4	$(a \wedge b) \vee (a \wedge 1)$
H3	$a \wedge (1 \vee b)$
R6a	$a \wedge (1)$
R6b	a ■

Aufgabe 3.2: Normalformen und Entwicklungssatz

A) Eine Schaltfunktion $Y(a, b)$ habe genau zwei Eingänge. Wie hoch ist die Anzahl aller theoretisch möglichen Schaltfunktionen mit dieser Eigenschaft? **1**

Es gibt $2^{(2^n)}$ 16 mögliche Schaltfunktionen mit 2 Eingängen.

B) Worin unterscheiden sich die Disjunktiven Normalform (DNF) und die Konjunktive Normalform (KNF)? Wie werden die Terme in den beiden Normalformen jeweils genannt und was Repräsentieren sie? **1**

Die DNF besteht aus einer Disjunktion sogenannter Minterme, wobei jeder Term zu genau einer Einsstelle gehört. Die KNF ist dagegen definiert als eine Konjunktion von Maxtermen, von denen jeder zu genau einer Nullstelle gehört.

C) Gegeben sei die nachfolgende Wahrheitstabelle der Logikfunktion $F(a, b, c)$. Bestimmen Sie die Konjunktive Normalform (KNF) der Funktion. **2**

a	b	c	F(a, b, c)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$F(a, b, c) = (a \vee b \vee c) \wedge (a \vee b \vee \overline{c}) \wedge (a \vee \overline{b} \vee \overline{c}) \wedge (\overline{a} \vee \overline{b} \vee \overline{c})$

D) Es sei eine weitere Logikfunktion $G(c,b,a) = a(c \vee b(c \vee a))$ gegeben. Wenden Sie den Entwicklungssatz der Schaltalgebra auf diese Funktion an. Entwickeln sie die Funktion sukzessive nach den einzelnen Variablen und vervollständigen Sie die nachfolgende Tabelle mit den Ergebnissen.

3

Entwicklung von $G(c,b,a)$ nach c:	
$G(0,b,a) =$ ab	$G(1,b,a) =$ a
Entwicklung von $G(0,b,a)$ nach b:	**Entwicklung von $G(1,b,a)$ nach b:**
$G(0,0,a) =$ 0	$G(1,0,a) =$ a
$G(0,1,a) =$ a	$G(1,1,a) =$ a
Entwicklung von $G(0,0,a)$ nach a:	**Entwicklung von $G(0,1,a)$ nach a:**
$G(0,0,0) =$ 0	$G(0,1,0) =$ 0
$G(0,0,1) =$ 0	$G(0,1,1) =$ 1
Entwicklung von $G(1,0,a)$ nach a:	**Entwicklung von $G(1,1,a)$ nach a:**
$G(1,0,0) =$ 0	$G(1,1,0) =$ 0
$G(1,0,1) =$ 1	$G(1,1,1) =$ 1

16

Aufgabe 4: Minimierung

Aufgabe 4.1: Symmetriediagramme

y					e	e	e	e	
			a	a		a	a		
	0_{0}	0_{1}	1_{5}	1_{4}	1_{20}	1_{21}	0_{17}	0_{16}	
b	1_{2}	1_{3}	0_{7}	1_{6}	0_{22}	0_{23}	1_{19}	1_{18}	
b	1_{10}	1_{11}	0_{15}	1_{14}	0_{30}	0_{31}	1_{27}	1_{26}	d
	0_{8}	0_{9}	1_{13}	1_{12}	0_{28}	1_{29}	0_{25}	0_{24}	d
			c	c	c	c			

Abbildung 4.1: Symmetriediagramm einer Schaltfunktion

A) Ist die Funktion in Abbildung 4.1 vollständig definiert? Begründen Sie ihre Antwort!

1

Ja, sie ist vollständig definiert da sie keine don't care Stellen enthält.

B) Bestimmen Sie nun für die Schaltfunktion aus Abbildung 4.1 eine **disjunktive** Minimalform (DMF).

4

DMF: $y = (b \wedge \bar{c}) \vee (\bar{a} \wedge c \wedge \bar{e}) \vee (a \wedge \bar{b} \wedge c) \vee (\bar{b} \wedge c \wedge \bar{d})$

C) Bestimmen Sie nun für die Schaltfunktion aus Abbildung 4.1 eine **konjunktive** Minimalform (KMF).

4

DMF: $y = (b \vee c) \wedge (\bar{a} \vee \bar{b} \vee \bar{c}) \wedge (\bar{b} \vee \bar{c} \vee \bar{e})$
Sowie einer der beiden folgenden Terme:
$(a \vee \bar{c} \vee \bar{d} \vee \bar{e})$ oder $(a \vee b \vee \bar{d} \vee \bar{e})$

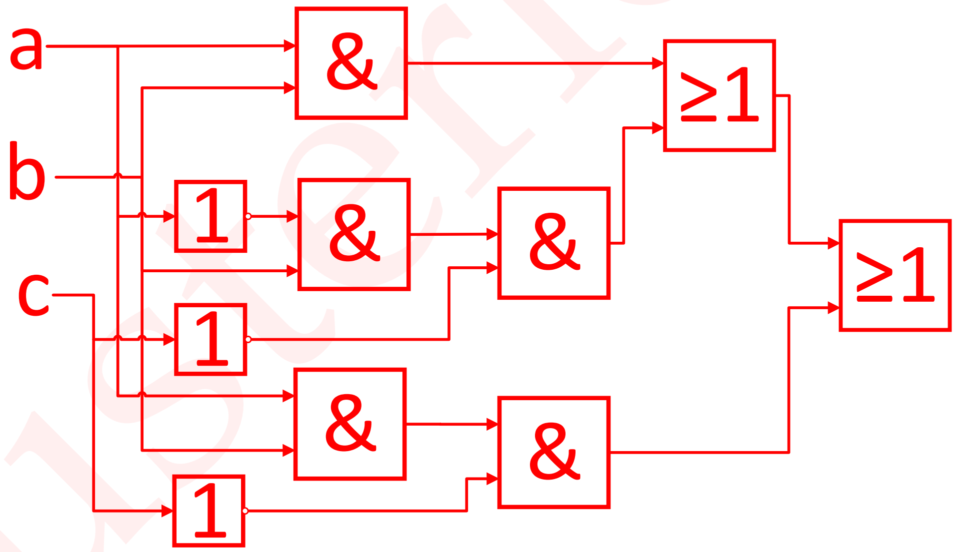
D) Wie setzt sich das Bewertungskriterium für Schaltfunktionen zusammen? Benennen Sie die Bestandteile und geben Sie den Wert für ihre ermittelte DMF aus Aufgabenteil B) an. **2**

Bewertungskriterium = Anzahl Literale + Anzahl Terme
= 11 Literale + 4 Terme
= 15

E) Gegeben sei die Schaltfunktion f(a,b,c): **2**

$(a \wedge b) \vee (\overline{a} \wedge b \wedge \overline{c}) \vee (a \wedge b \wedge \overline{c})$

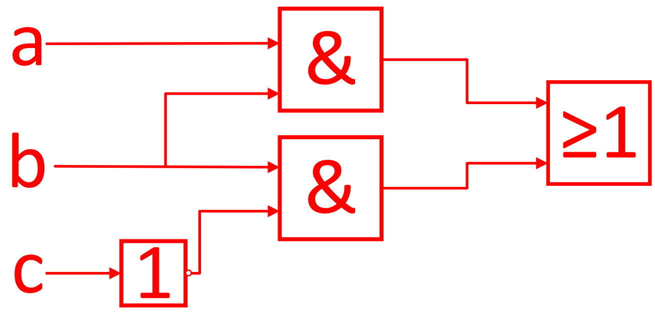
Zeichnen Sie für die angegebene Funktion das zugehörige Schaltbild unter Verwendung der folgenden Elemente: Inverter, UND-Gatter (zwei Eingänge), ODER-Gatter (zwei Eingänge). Die Eingangsliterale sollen dabei nur einmal und nicht bereits invertiert vorkommen. Die Funktion soll zunächst nicht minimiert werden und muss die Schaltfunktion exakt wie angegeben abbilden.



F) Bestimmen Sie die DMF , durch ein von Ihnen gewähltes Verfahren, der Funktion aus der vorigen Aufgabe E). Zeichnen Sie anschließend das Schaltbild für die von Ihnen minimierte Funktion unter Verwendung der gleichen Elemente und Einschränkungen wie in der vorigen Aufgabe. Ihr Vorgehen und der Lösungsweg bei der Minimierung müssen in Ihrer Lösung erkennbar sein.

3

Lösung z.B. über Umformung, KV Diagramm, Wahrheitstabelle. Die minimierte Form lautet: $(a \wedge b) \vee (b \wedge \bar{c})$



Aufgabe 5: Mengen, Relationen und Graphen

16

Aufgabe 5.1: Mengen

Gegeben seien eine Grundmenge G sowie zwei Mengen A und B:

$$G = \{a, b, c, 0, 2, 3, 5, 6, 7\}$$
$$A = \left\{x | x \in G \; und \left(x = c \text{ oder x ist Primzahl oder x ist positive Zahl und } (x-3)^2 = 4\right)\right\}$$
$$B = \{a, 0, 2, 6\} \tag{5.1}$$

A) Wie lauten die Elemente x der Menge A? **1**

c, 2, 3, 5, 7

B) Wie viele Elemente hat die Potenzmenge P(B). Geben Sie ihre Elemente an. **3**

$|P(B)| = 2^4 = 16$
P(B) = {{}, {a}, {0}, {2}, {6}, {a,0}, {a,2}, {a,6}, {0,2}, {0,6}, {2,6}, {a,0,2}, {a,0,6}, {a,2,6}, {0,2,6}, {a,0,2,6}}

C) Wie ist das kartesische Produkt zweier Mengen S und T definiert? Bestimmen Sie die Mächtigkeit (auch Kardinalität genannt) des kartesischen Produktes A x B x G. **2**

1te Def.: Das kartesische Produkt SxT (gesprochen: "S kreuz T") zweier Mengen S und T ist die Menge aller **geordneten** Paare (s,t) mit $s \in S$ und $t \in T$.
2te Def.: $S \times T := (s, t) | s \in S, t \in T$

$|A \times B \times G| = |A| \cdot |B| \cdot |G| = 5 \cdot 4 \cdot 9 = 180$

D) Bilden sie bezüglich der Grundmenge G das Komplement $C_G(B)$. **1**

$C_G(B) = \{b, c, 3, 5, 7\}$

Aufgabe 5.2: Relationen und Graphen

Durch den in Abb. 5.1 dargestellten Graphen sei die Relation γ zwischen zwei Mengen Y und Z definiert.

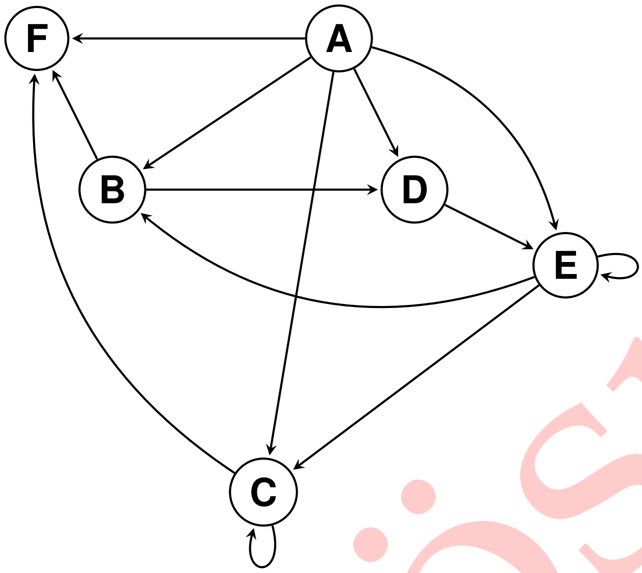


Abbildung 5.1: Graph G für die Dartellung einer Relation γ

A) Beurteilen Sie die Relation γ hinsichtlich der Eigenschaften: Reflexivität, Symmetrie, Antisymmetrie, Transitivität. Besitzt sie jeweils diese Eigenschaft oder nicht? Begründen Sie Ihre Antworten. **4**

Reflexivität: nicht reflexiv, da Schleifen nur an manchen Knoten: z. B. $\overline{A\gamma A}$
Symmetrie: nicht symmetrisch, da keine Antiparallele Kanten: z. B. $A\gamma D$, $\overline{D\gamma A}$
Antisymmetrie: antisymmetrisch, da überall, wo $x\gamma y \wedge y\gamma x$ gilt $x = y(E, C)$ ist.
Transitivität: nicht transitiv, da z. B. $E\gamma C$ und $C\gamma F$, aber $\overline{E\gamma F}$

B) Ist der Graph in Abbildung 5.1 ein Baum? Begründen Sie Ihre Antwort. **1**

Nein, ist nicht zyklenfrei

Gegeben sei der Graph H in Abbildung 5.2.

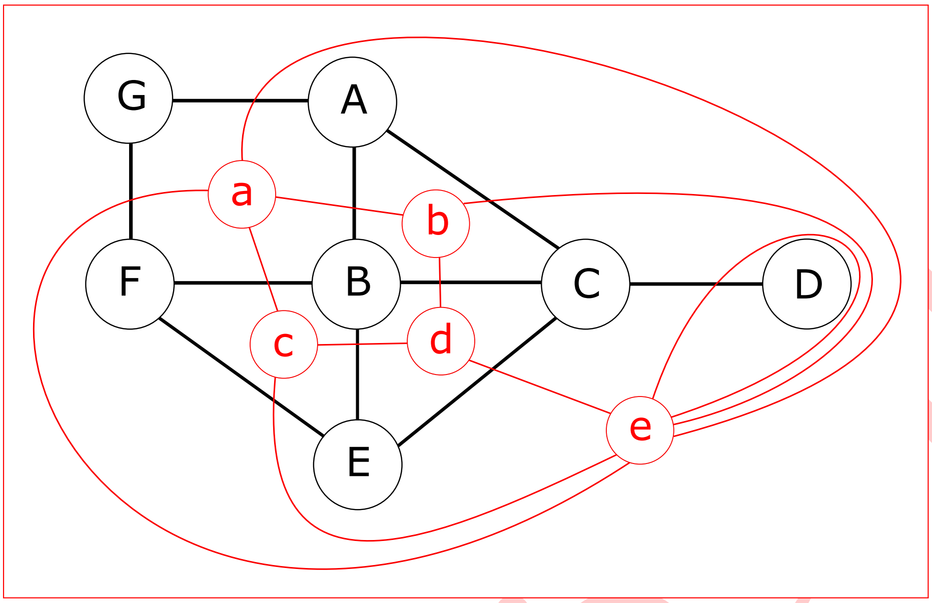


Abbildung 5.2: Graph H

C) Welche Bedingung ist für einen bestimmten Graphen notwendig für die Konstruktion eines entsprechenden dualen Graphen? Ist diese Bedingung hinreichend? **1**

Notwendige Bedingung: Planarität des Graphen.
Sie ist hinreichend.

D) Konstruieren Sie zum Graphen H einen dualen Graphen. Zeichnen Sie direkt in Abbildung 5.2. Kennzeichnen Sie die Knoten mit den Buchstaben {a, b, c, ...}. **3**

Aufgabe 6: Optimale Codes

13

Die Abbildung 6.1 zeigt ein 120x120 pixel Bild von Erdbeeren. Jeder Pixel hat acht mögliche Graustufen. Die verwendete Codierung in der Abbildung ist binär und mit gleicher Länge.

Abbildung 6.1: Erdbeeren. Bild 120 x 120 p

A) Wie viele Bits pro Pixel benötigt die originale Kodierung des Bildes? 1

Binär, gleiche Länge, 8 Graustufe –> 3 Bits

B) Komprimieren Sie das Bild mithilfe des Huffmann-Codes. Verwenden Sie dafür die in der Tabelle 6.1 gezeigte Auftrittswahrscheinlichkeit jeder Graustufe. **Verwenden Sie für die Herstellung des Baumes die nächste Seite und füllen Sie anschließend die Tabelle aus.** Konventionen: 6

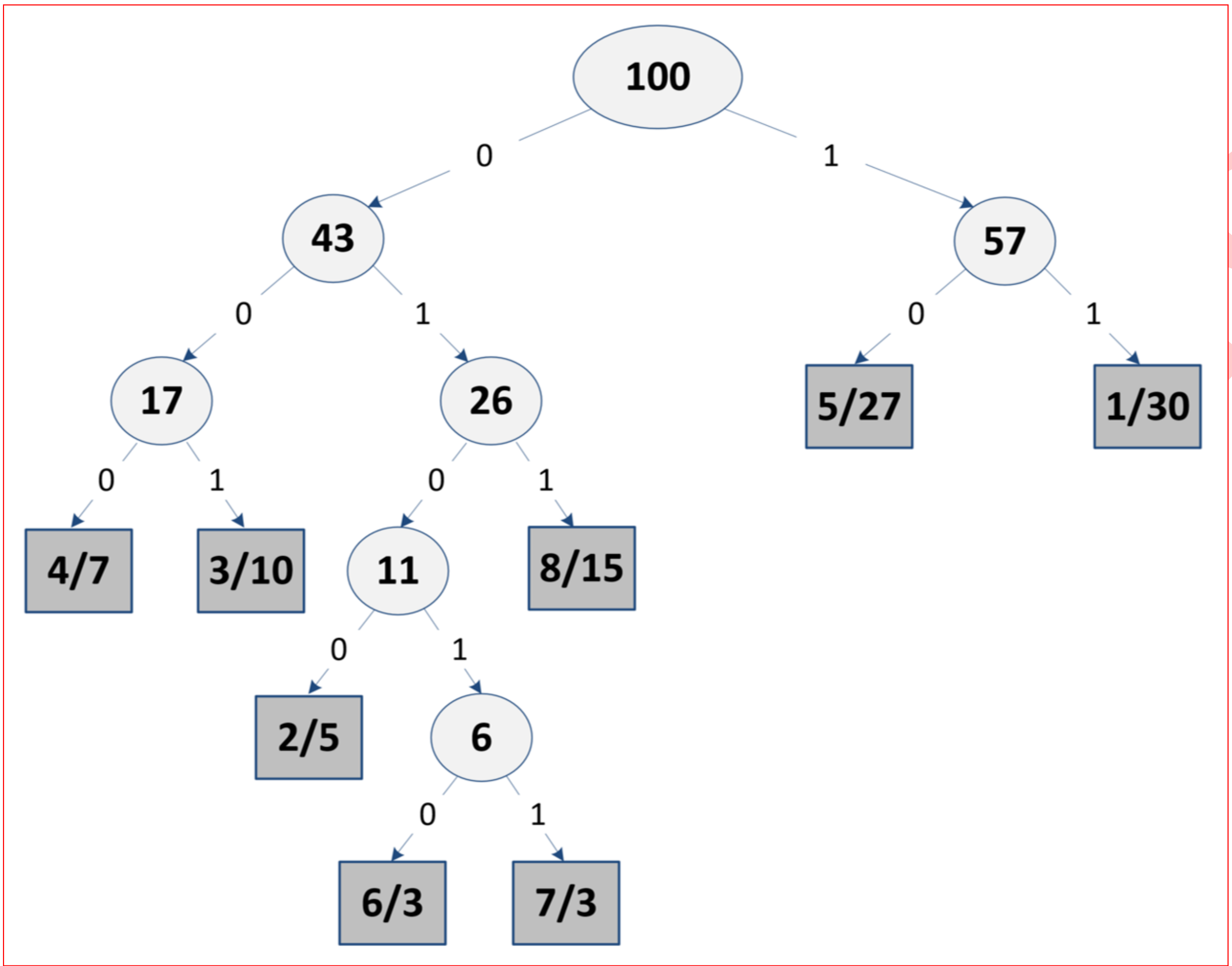
- Der Lösungsweg muss nachvollziehbar sein und der vollständige Codierbaum angegeben werden!
- Sortieren Sie die Elemente entsprechend den Auftrittshäufigkeiten **aufsteigend von links nach rechts**. Falls unterschiedliche Knoten dieselbe Auftrittshäufigkeiten haben, sortieren Sie diese numerisch ebenfalls von links nach rechts.
- Weisen Sie den linken Ästen des entstehenden Baumes die „0“ zu, den rechten Ästen die „1“.

Graustufe	Wahrscheinlichkeit	Code
1	0,30	11
2	0,05	0100
3	0,10	001
4	0,07	000
5	0,27	10
6	0,03	01010
7	0,03	01011
8	0,15	011

Tabelle 6.1: Auftrittshäufigkeiten jeder Graustufe der Abb. 6.1

Antwort Frage (B):

Antwort Frage (B):



C) Geben Sie die Formel zur Berechnung der mittleren Codewortlänge Ihrer Huffmann-Codierung und berechnen Sie sie entsprechend. **2**

$\bar{m} = \sum_{i=1}^{n} m(x_i)p(x_i) = 2 \cdot 0,3 + 4 \cdot 0,05 + 3 \cdot 0,1 + 3 \cdot 0,07 + 2 \cdot 0,27 + 5 \cdot 0,03 + 5 \cdot 0,03 + 3 \cdot 0,15 = 2,6\,bits$

D) Was ist die erreichte Kompressionsrate Ihres Huffman-Codes? **1**

$3\,bits/2,6\,bits$

E) Dekodieren Sie die folgende binäre Reihe anhand vom in der Tabelle 6.2 gezeigten Code eines Kodierbaumes. Was ist der Grund, dass die Reihe dekodierbar ist? **2**

11011100011011

Präfixfreieheit (kein Codewort ist Präfix eines anderen Codewortes).
11 011 10 00 11 011 = D,E,C,A,D,E

Buchstaben	**Code**
A	00
B	010
C	10
D	11
E	011

Tabelle 6.2: Code Frage (E)

F) In der Praxis kennt man die Auftrittswahrscheinlichkeiten der Farben der Bilder normalerweise nicht vor der Kodierbaumerzeugung. Schreiben Sie eine Möglichkeit zur Verbesserung des Huffman Verfahrens in diesem Fall. **1**

Iterative Erzeugung der Bäumen, wie den Fall des adaptiven Huffman Codes.
(Fast) Discrete Cosine Transform, wie beim JPEG.

18

Aufgabe 7: Automaten

Aufgabe 7.1: Analyse eines Automaten

Gegenstand dieser Aufgabe ist ein Automat mit den Zuständen Z_0, Z_1, Z_2 und Z_3 sowie der Eingabe X. Abbildung 7.1 zeigt ein unvollständiges Ablaufdiagramm dieses Automaten.

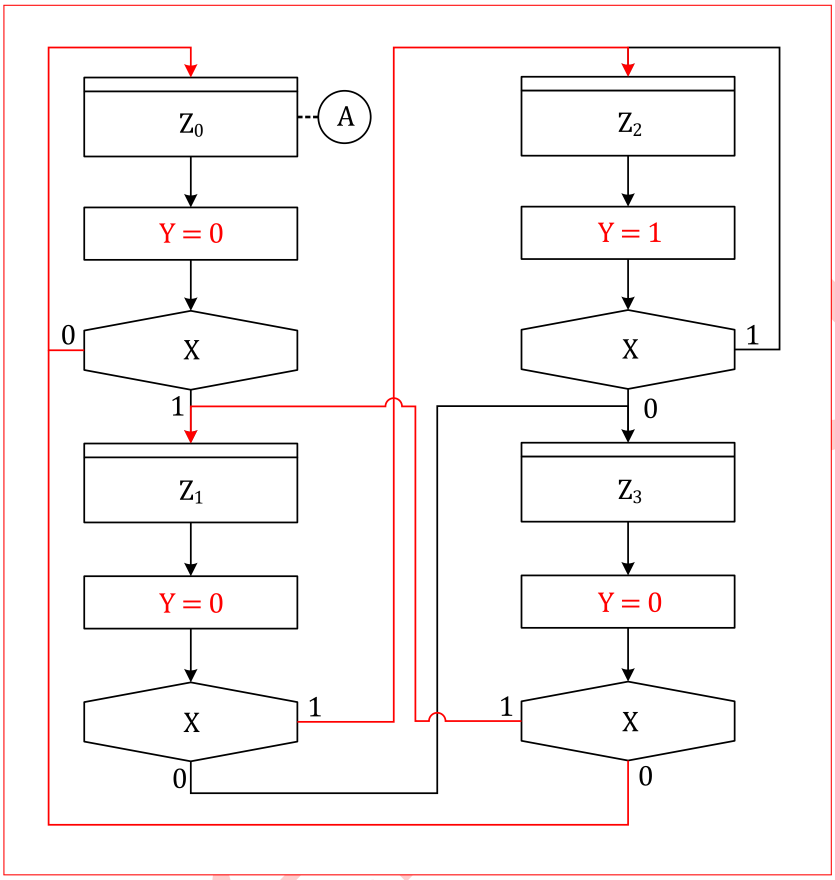


Abbildung 7.1: Ablaufdiagramm des betrachteten Automaten

A) Das unvollständige Ablaufdiagramm umfasst vier Zustandsübergänge. Vervollständigen Sie nun Tabelle 7.1, indem Sie diese Zustandsübergänge betrachten und die jeweiligen Folgezustände in die dafür vorgesehenen Zellen der Tabelle eintragen.

2

Zustand	Eingabe	Folgezustand
S^v	$E^v = X$	S^{v+1}
Z_0	1	Z_1
Z_1	0	Z_3
Z_2	0	Z_3
	1	Z_2

Tabelle 7.1: Zustandsübergänge aus dem Ablaufdiagramm

B) Die in Abbildung 7.1, dem unvollständigen Ablaufdiagramm, fehlenden Zustandsübergänge sind in Tabelle 7.2 aufgeführt. Zeichnen Sie diese Zustandsübergänge in das Ablaufdiagramm ein.

2

Zustand	**Eingabe**	**Folgezustand**
S^v	$E^v = X$	S^{v+1}
Z_0	0	Z_0
Z_1	1	Z_2
Z_3	0	Z_0
	1	Z_1

Tabelle 7.2: Zustandsübergänge zur Übertragung ins Ablaufdiagramm

Der nachfolgende Aufgabenteil bezieht sich auf das Ablaufdiagramm nach Erweiterung um die fehlenden Zustandsübergänge. Bei dem Zustandsautomaten, den es beschreibt, handelt es sich um ein **2-Bit-Schieberegister**. X kennzeichnet den Wert, der zur Taktflanke ins Schieberegister übernommen wird. Die betrachteten Zustände (Z_0, Z_1, Z_2 und Z_3) repräsentieren je einen der vier möglichen Schieberegisterinhalte. Initial, d. h. im Anfangszustand Z_0, enthält das Schieberegister zwei ‚0'-Bits.

C) Nun sollen die Zustände des Automaten gemäß des Schieberegisterinhalts kodiert werden. Vervollständigen Sie Tabelle 7.3, indem Sie die Spalten für S_0^v und S_1^v ausfüllen. Für jeden Zustand S^v stellt S_0^v das zuletzt ins Schieberegister übernommene Bit dar. S_1^v ist dagegen das im betrachteten Zustand älteste Bit des Schieberegisters.

2

Zustand	**Bit 1**	**Bit 0**
S^v	S_1^v	S_0^v
Z_0	0	0
Z_1	0	1
Z_2	1	1
Z_3	1	0

Tabelle 7.3: Zustandskodierung des betrachteten Automaten

In den folgenden Aufgabenteilen soll zusätzlich die Ausgabe Y des Automaten betrachtet werden. Für diese gilt genau dann $Y = 1$, wenn der Automat zwei ‚1'-Bits enthält. Andernfalls, d. h. bei mindestens einem enthaltenen ‚0'-Bit, gilt $Y = 0$.

D) Vervollständigen Sie das Ablaufdiagramm in Abbildung 7.1 nun endgültig, indem Sie die bisher leeren Ausgabefelder mit den spezifizierten Werten befüllen. **1**

Gemäß der Kodierung in Tabelle 7.3 enthält der Automat genau dann die beiden ‚1'-Bits, wenn er sich in Z_2 befindet. Für diesen Zustand gilt also $Y = 1$. Für die übrigen Zustände gilt $Y = 0$. Dies ist in Abbildung 7.1 eingezeichnet.

E) Handelt es sich beim vervollständigten Automaten in Abbildung 7.1 um einen Mealy- oder um einen Moore-Automaten? Geben Sie eine kurze Begründung für Ihre Entscheidung an. **1**

Da die Ausgabe in der dargestellten Form nur vom aktiven Zustand abhängt, handelt es sich um einen Moore-Automaten: $A_h^v = \lambda\left(S_h^v\right)$.

F) Warum ist es nicht möglich, den betrachteten Automaten unter Beibehaltung der Funktionalität in einen Medwedew-Automaten zu transformieren? Begründen Sie. **1**

Es müssen drei unterschiedliche Zustände mit der gleichen Ausgabe existieren (Z_0, Z_1 und Z_3). Da die Ausgabe des Medwedew-Automaten dem aktiven Zustand entspricht, wäre die beschriebene Funktionalität nicht realisierbar.

Aufgabe 7.2: Realisierung von Automaten

6

A) Tabelle 7.4 zeigt die Ablauftabelle eines Automaten. Dieser soll mit einem flankengesteuerten RS-Fliplop für das erste Bit (S_0^v), einem T-Flipflop für das zweite Bit (S_1^v) und einem D-Flipflop für das dritte Bit (S_2^v) realisiert werden. d_2 bezeichnet den D-Eingang des D-Flipflops. t_1 bezeichnet den T-Eingang des T-Flipflops. s_0 und r_0 beziehen sich auf die S- und R-Eingänge des RS-Flipflops. Vervollständigen Sie die Ablauftabelle, indem Sie die fehlenden Ansteuerungen für d_2, t_1, s_0 und r_0 angeben. Nutzen Sie Don't-Care-Stellen, wann immer es möglich ist.

Zustand	Eingabe	Folgezustand	Flipflop-Ansteuerung			
$S^v = (S_2^v, S_1^v, S_0^v)$	E^v	S^{v+1}	d_2	t_1	s_0	r_0
0, 0, 0	0	0, 0, 0	0	0	0	-
	1	0, 0, 1	0	0	1	0
0, 0, 1	0	0, 0, 0	0	0	0	1
	1	0, 1, 0	0	1	0	1
0, 1, 0	0	0, 0, 0	0	1	0	-
	1	0, 1, 1	0	0	1	0
0, 1, 1	0	0, 0, 0	0	1	0	1
	1	1, 0, 0	1	1	0	1
1, 0, 0	0	0, 0, 0	0	0	0	-
	1	1, 0, 1	1	0	1	0
1, 0, 1	0	0, 0, 0	0	0	0	1
	1	1, 1, 0	1	1	0	1
1, 1, 0	0	0, 0, 0	0	1	0	-
	1	1, 1, 1	1	0	1	0
1, 1, 1	0	0, 0, 0	0	1	0	1
	1	1, 1, 1	1	0	-	0

Tabelle 7.4: Ablauftabelle eines Zustandsautomaten

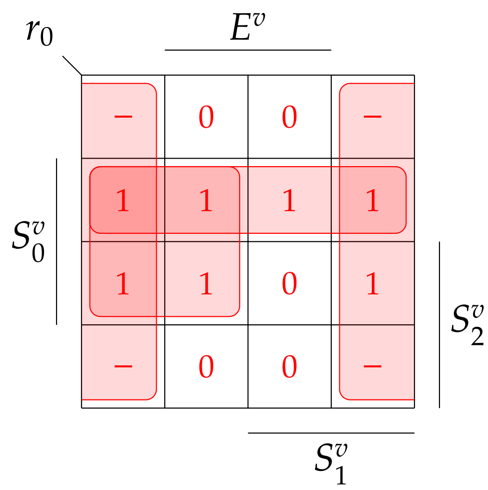


Abbildung 7.2: Symmetriediagramm

B) Ziel ist es nun, die Ansteuerfunktion für den R-Eingang zu minimieren. Tragen Sie die Werte, die Sie im vorausgehenden Aufgabenteil für r_0 erhalten haben, in das Symmetriediagramm in Abbildung 7.2 ein. Bestimmen Sie auf dessen Basis eine disjunktive Minimalform (DMF) für die Ansteuerung des r_0-Eingangs und geben Sie diese an. **1**

$r_0 = \overline{E^v} \vee \left(S_0^v \wedge \overline{S_1^v}\right) \vee \left(S_0^v \wedge \overline{S_2^v}\right)$

C) Ergänzen Sie die Schaltung in Abbildung 7.3 so, dass ein JK-Flipflop entsteht. Nutzen Sie dafür das flankengesteuerte RS-Flipflop und so wenig zusätzliche Gatter, wie möglich. **2**

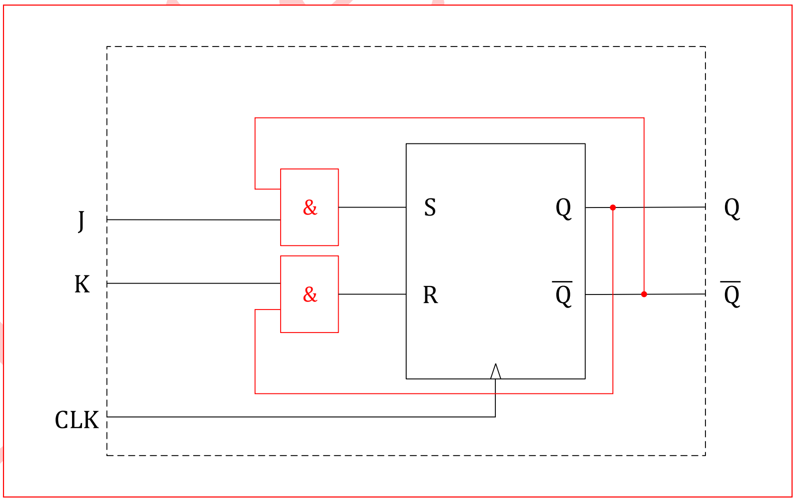


Abbildung 7.3: RS-Flipflop zur Erzeugung eines JK-Flipflops

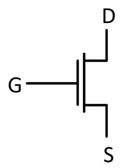
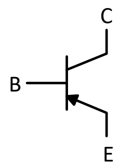
Aufgabe 8: CMOS und Gatter

15

Hinweis: Verwenden Sie für diese Aufgabe eine **positive Logik**, der CMOS-Pegel VCC entspricht einer logischen '1'.

Aufgabe 8.1: Transistortypen

A) In der untenstehenden Tabelle sind die Schaltbilder zweier unterschiedlicher Transistoren abgebildet. Geben Sie in der zweiten Spalte den grundlegenden Transistortypen und die konkrete Variante an. 1

Schaltbild	Transistortyp
	n-Kanal MOS (NMOS) Feldeffekttransistor (FET)
	PNP Bipolartransistor

B) Nennen Sie die volle Bezeichnung des G Anschlusses von MOSFETs und erläutern Sie kurz dessen Funktion. 1

G: Gate
Funktion: Schalter / steuert Schalter / schaltet Kanal / oder ähnliche Formulierung

Aufgabe 8.2: Fehleranalyse in CMOS-Schaltungen

A) Nennen Sie die Namen und formalen Bedingungen der zwei Kriterien, die für ein wohldefiniertes CMOS-Schaltnetz gelten müssen. **1**

Kurzschlussfreiheit: $v_1 \cdot v_2 = 0$
Vollständigkeit: $v_1 + v_2 = 1$ (auch OK: F, G als Variablen)

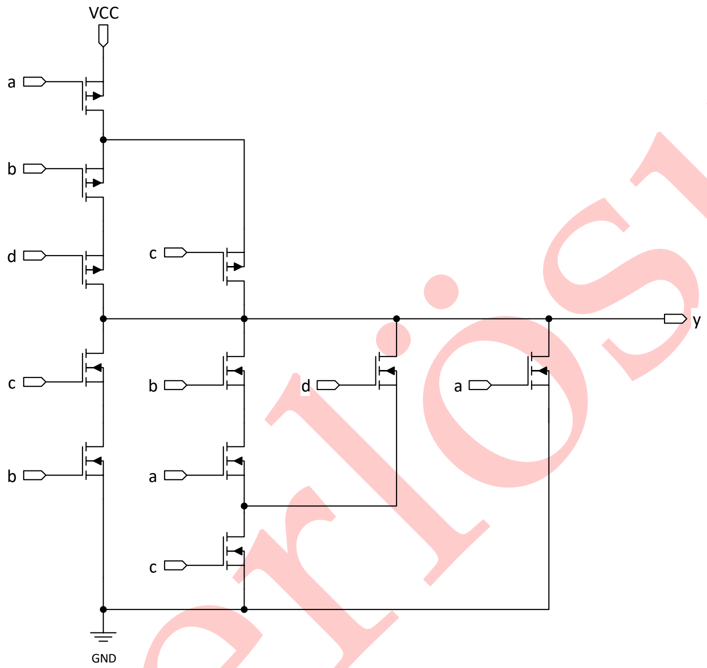


Abbildung 8.1: CMOS-Schaltung

B) In Abbildung 8.1 ist eine CMOS-Schaltung dargestellt. Geben Sie die Pull-Up-Funktion F und die Pull-Down-Funktion G als Disjunktion von Konjunktionstermen an. Prüfen Sie anhand der entsprechenden Bedingung aus der vorherigen Teilaufgabe, ob es Eingangskombinationen gibt, für die sowohl das Pull-Up als auch das Pull-Down Netz gleichzeitig durchschalten. **3**

$F = \overline{a} \wedge \left(\overline{b}\,\overline{d} \vee \overline{c}\right) = \overline{a}\overline{b}\,\overline{d} \vee \overline{a}\,\overline{c}$
$G = bc \vee c\,(ab \vee d) \vee a = bc \vee cab \vee cd \vee a$

Kurzschlüsse:
$F \cdot G = 0$
$F \cdot G = \left(\overline{a}\overline{b}\,\overline{d} \vee \overline{a}\,\overline{c}\right) \wedge (bc \vee cab \vee cd \vee a)$
$= \overline{a}\overline{b}\,\overline{d}bc \vee \overline{a}\overline{b}\,\overline{d}cab \vee \overline{a}\overline{b}\,\overline{d}cd \vee \overline{a}\overline{b}\,\overline{d}a \vee \overline{a}\,\overline{c}bc \vee \overline{a}\,\overline{c}cab \vee \overline{a}\,\overline{c}cd \vee \overline{a}\,\overline{c}a$
$= 0$
Keine Kurzschlüsse.

C) Minimierte Sie die DNF der Pull-Down Funktion und geben Sie die DMF an. Zeigen Sie anschließend, dass die Schaltung aus Abbildung 8.1 wohldefiniert ist. **3**

Minimierung

$G = bc \vee cab \vee cd \vee a = bc \vee cd \vee a = G_{min}$

Wohldefiniert:

$\overline{F} = \underline{G}$

$\overline{F} = \overline{a}\,\overline{b}\,\overline{d} \vee \overline{a}\,\overline{c}$

$= (a \vee b \vee d) \wedge (a \vee c)$

$= a \vee ac \vee ba \vee bc \vee da \vee dc$

$= a \vee bc \vee dc$

$= bc \vee cd \vee a$

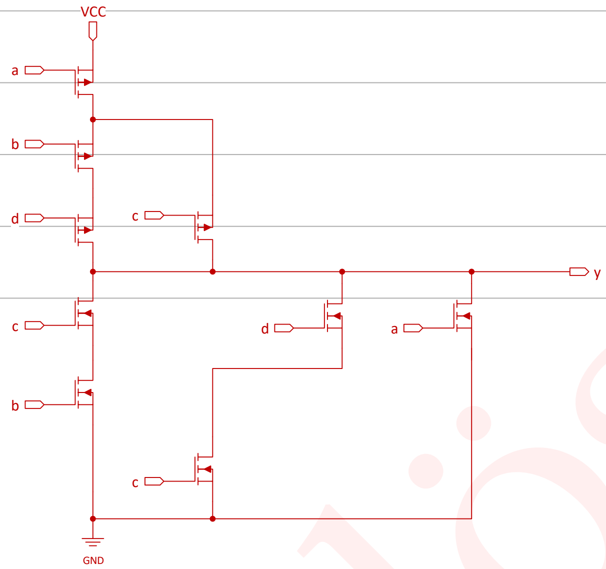
$= G_{min}$

Die Schaltung ist wohldefiniert

D) Zeichnen Sie die DMF aus Aufgabe C) **entsprechende** CMOS Schaltung, inklusive Pull-Up Netzwerk. Begründen Sie mit dem Ergebnis aus Aufgabe C) und **ohne Rechnung**, dass die hier gezeichnete Schaltung wohldefiniert ist.

2

Zeichnung:



Wohldefiniert:
Die minimierte Pull-Down Funktion ist funktional äquivalent zur ursprünglichen Pull-Down Funktion. Da bereits gezeigt wurde, dass die ursprüngliche Schaltung wohldefiniert ist, muss die neue Schaltung ebenfalls wohldefiniert sein:
$G = G_{min}; F = \overline{G} = \overline{G_{min}}$

E) Auch diese Schaltung aus Aufgabe D) ist bezüglich der Anzahl der verwendeten Transistoren nicht minimal. Weshalb kann die CMOS Schaltung bezüglich der Anzahl der Transistoren noch weiter minimiert werden? **1**

Weitere Minimierung:
Im PDN kann auch mehrstufige Logik realisiert werden, die nicht der zweistufigen Form der DNF entspricht. Deshalb ist eine minimierte DNF nicht zwingend die minimale Darstellung bezüglich der Anzahl der Transistoren. Beispiel:
$G_{Tmin} = (c \land (b \lor d)) \lor a$

Aufgabe 8.3: Analyse einer CMOS-Logikschaltung

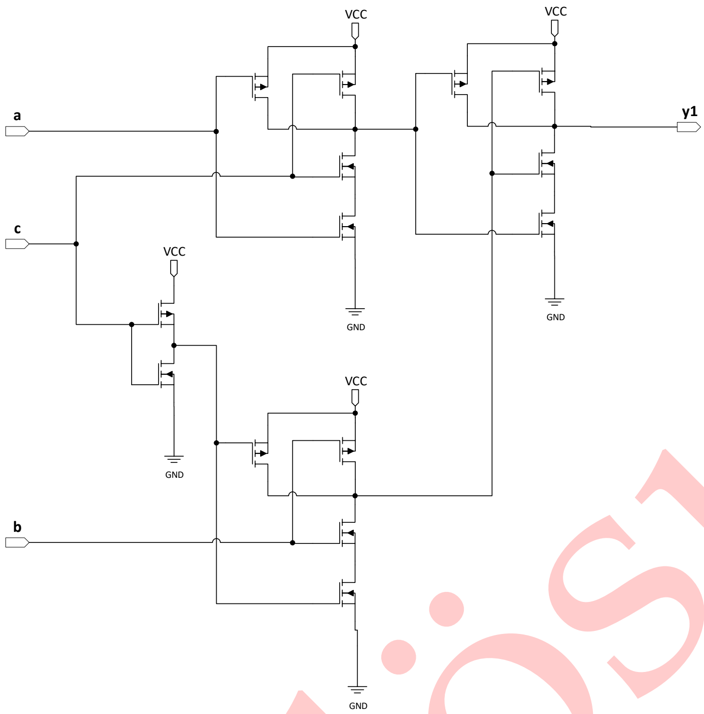
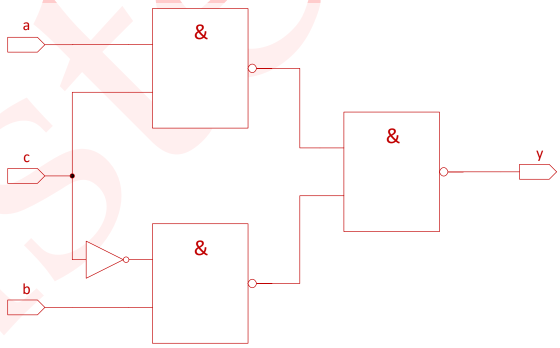


Abbildung 8.2: CMOS-Schaltung aus mehreren Gattern

A) Gegeben sei die CMOS-Schaltung aus Abbildung 8.2. Rekonstruieren Sie daraus eine äquivalente Schaltung aus Logikgattern. Verwenden Sie ausschließlich NAND, NOR, NOT Gatter. **2**



B) Welche aus der Vorlesung bekannte Funktion realisiert die Schaltung? **1**

Multiplexer

Zusatzblatt zu Aufgabe $\boxed{}$:

Formelblatt Digitaltechnik

Huntingtonschen Axiome für alle $a,b,\mathrm{I},\mathrm{O} \in K$; $\bar{a} = k \in K$

Abgeschlossenheit:	(H1)	$a \top b \in K$	$a \perp b \in K$
Kommutativgesetz:	(H2)	$a \top b = b \top a$	$a \perp b = b \perp a$
Distributivgesetz:	(H3)	$(a \top b) \perp c = (a \perp c) \top (b \perp c)$	$(a \perp b) \top c = (a \top c) \perp (b \top c)$
Neutrales Element:	(H4)	$\mathrm{I} \top a = a$	$\mathrm{O} \perp a = a$
Komplement:	(H5)	$a \top k = \mathrm{O}$	$a \perp k = \mathrm{I}$

Abgeleitete Regeln

R1a:	$\bar{0} = 1$	R1b:	$\bar{1} = 0$
R2a:	$0 \vee 0 = 0$	R2b:	$1 \,\&\, 1 = 1$
R3a:	$1 \vee 1 = 1$	R3b:	$0 \,\&\, 0 = 0$
R4a:	$1 \vee 0 = 1$	R4b:	$1 \,\&\, 0 = 0$
R5a:	$a \vee 0 = a$	R5b:	$a \,\&\, 0 = 0$
R6a:	$a \vee 1 = 1$	R6b:	$a \,\&\, 1 = a$
R7a:	$a \vee a = a$	R7b:	$a \,\&\, a = a$
R8a:	$a \vee \bar{a} = 1$	R8b:	$a \,\&\, \bar{a} = 0$
R9:	$\overline{(\bar{a})} = \bar{\bar{a}} = a$		

Assoziative Gesetze:

R10a: $(a \vee b) \vee c = a \vee (b \vee c) = a \vee b \vee c$ R10b: $(a \,\&\, b) \,\&\, c = a \,\&\, (b \,\&\, c) = a \,\&\, b \,\&\, c$

Absorptionsgesetze:

R11a: $(a \vee b) \,\&\, a = a$ R11b: $(a \,\&\, b) \vee a = a$

De Morgan:

R12a: $\overline{(a \vee b)} = \bar{a} \,\&\, \bar{b}$ R12b: $\overline{(a \,\&\, b)} = \bar{a} \vee \bar{b}$

Umwandlung DNF $\leftrightarrow$ KNF

$$OR(AND(l_i)) \xleftrightarrow{R9} OR\left(AND(\bar{\bar{l_i}})\right) \xleftrightarrow{R12a} OR\left(\overline{OR(\bar{l_i})}\right) \xleftrightarrow{R12b} \overline{AND\left(OR(\bar{l_i})\right)} \xleftrightarrow{H3}$$

$$\overline{OR\left(AND(\bar{l_k})\right)} \xleftrightarrow{R12a} AND\left(\overline{AND(\bar{l_k})}\right) \xleftrightarrow{R12b} AND\left(OR(\bar{\bar{l_k}})\right) \xleftrightarrow{R9} AND\left(OR(l_k)\right)$$

Weitere Funktionen

Implikation $a \to b = \bar{a} + b$

Äquivalenz $a \equiv b = a \cdot b + \bar{a} \cdot \bar{b}$

Inklusion $f \le g \Rightarrow f \cdot \bar{g} = 0$

Transitivität der Inklusion

$(f \le g) \cdot (g \le h) \Rightarrow f \le h$

Tautologie $f = g \Leftrightarrow f \equiv g = 1$

$f \le g \Leftrightarrow \bar{f} + g = 1$

XOR-Regeln $x \oplus y = \bar{x} \cdot y + x \cdot \bar{y}$

$x + y = x \cdot y \oplus x \oplus y$

$x \cdot (y \oplus z) = x \cdot y \oplus x \cdot z$

$\bar{x} = x \oplus 1$, $x \oplus x = 0$, $x \oplus 0 = x$

Multiplexer $f = x \cdot a + \bar{x} \cdot b$

Entwicklungssatz

Boolescher Entwicklungssatz $f = x \cdot f_x + \bar{x} \cdot f_{\bar{x}}$

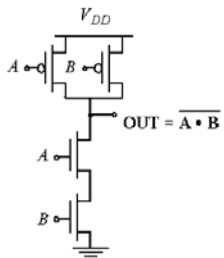
Dualer Entwicklungssatz $f = (\bar{x} + f_x) \cdot (x + f_{\bar{x}})$

CMOS Schaltungen

CMOS (wohl definiert) $v_1 = \overline{v_0}$

CMOS (kein Kurzschluss) $v_1 \cdot v_0 = 0$

CMOS (Vollständigkeit) $v_1 + v_0 = 1$



Addierer

Halbaddierer $s_i = a_i \oplus b_i$ $c_{i+1} = a_i \cdot b_i$

Volladdierer $s_i = a_i \oplus b_i \oplus c_i$

$c_{i+1} = a_i \cdot b_i + (a_i \oplus b_i) \cdot c_i$

Carry-Look-ahead $c_{i+1} = g_i + p_i \cdot c_i$

Generate $g_i = a_i \cdot b_i$

Propagate $p_i = a_i \oplus b_i$

Informationsgehalt

Informationsgehalt H_e eines Zeichens: $H_e = \mathrm{ld}\,\frac{1}{p}$

Informationsgehalt H einer Quelle: $H = \sum_{i=1}^{N} p(i) \cdot \mathrm{ld}\,\frac{1}{p(i)}$

mit der Auftrittswahrscheinlichkeit $p(i)$ und $\sum_{i=1}^{N} p(i) = 1$

a, b, x, y, z : boolesche Variablen l : Literal f, g : boolesche Funktionen v_1 : p-Netz v_0 : n-Netz

s : Summe/Zustand c : Carry i : Eingang δ : Transitionsfunktion λ : Ausgabefunktion

Codierung

Allgemeiner Aufbau einer polyadischen Zahl:

$$N = dn * Rn + \ldots + d1 * R1 + d0 * R0$$

mit *N* Zahl im Zahlensystem; *R* Basis; R^i Wertigkeit; d_i Ziffer der i-ten Stelle; *Z* Menge der Ziffer $d_i \in \{0, 1, 2, \ldots, R-1\}$

ASCII-Tabelle

LSB	MSB							
Binär	000	001	010	011	100	101	110	111
	Steuerzeichen			Großbuchstaben			Kleinbuchstaben	
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	„	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	´	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Anzahl Codewörter im (*k* aus m)-Code:

$$\binom{m}{k} = \frac{m!}{k!(m-k)!}$$

Korrigierbare Fehleranzahl bei ungerader HD:

$$F_k = \frac{(d-1)}{2}$$

Erkennbare Fehleranzahl: $F_e = d - 1$

Gleitkommadarstellung gemäß IEEE 754 Standard

Vorzeichen | Exponent | Mantisse

V									2^{-1}	2^{-2}	2^{-3}	2^{-4}	...	2^{-22}	2^{-23}

Bit 31 30 23 22 0

Exponent E	Mantisse M	Wert
255	≠0	ungültig (NAN)
255	0	$-1^V \cdot \infty$ (±unendlich)
0 < E < 255	**M**	**$-1^V \cdot 2^{E-127} \cdot (1,M)$**
0	[illegible]	$-1^V \cdot 2^{-126} \cdot (0,M)$
0	0	$-1^V \cdot 0$

Minimierung - Allgemeine Vorgehensweise:

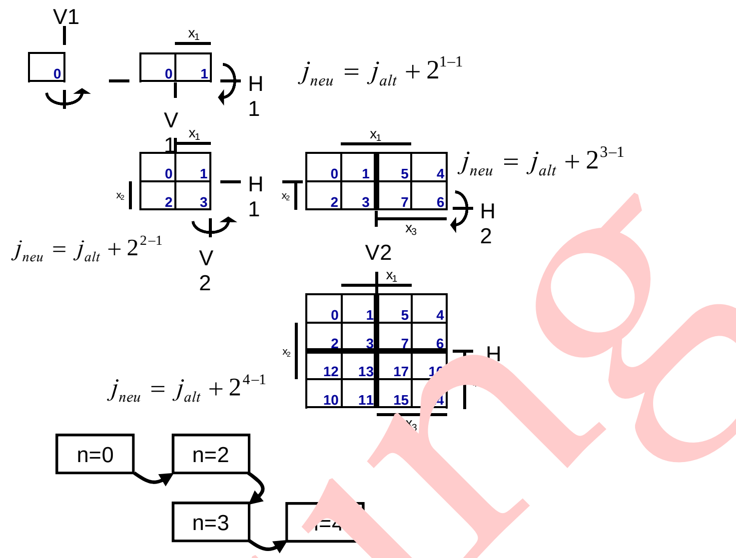
1) Kerne bestimmen und Streichen aller überdeckten Spalten (@ Einsstellen) ("leergewordene" Zeilen können auch gestrichen werden)

2) Spaltendominanzen finden und **dominierende Spalten streichen**

3) Zeilendominanzen finden und **dominierte Zeilen streichen**, nach Möglichkeit (-> **Kosten ci beachten**!)

4) Schritte **1-3 wiederholen**, bis **Überdeckungstabelle nicht reduzierbar** -> keine Kerne und Dominanzen mehr (ggf. noch zyklische Resttabelle auflösen)

Entwicklung eines Symmetriediagramms



FlipFlops (charakteristische Gleichungen)

RS-FlipFlop:

$q_k^{v+1} = S^v \vee (q_k \& \overline{R}^v)$

q^v	q^{v+1}	R	S
0	0	-	0
0	1	0	1
1	0	1	0
1	1	0	-

D-FlipFlop:

$q_k^{v+1} =$ [illegible]

q^v	q^{v+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

JK-FlipFlop:

$q^{v+1} = (\overline{K} \& q^v) \vee (J \& \overline{q}^v)$

q^v	q^{v+1}	K	J
0	0	-	0
0	1	-	1
1	0	1	-
1	1	0	-

T-FlipFlop:

$q^{v+1} = (T \& \overline{q}^v) \vee (\overline{T} \& q^v)$

q^v	q^{v+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Automaten

A_h^v Ausgangsvektor;

S_k^v Zustandsvektor;

E_g^v Eingangsvektor

Transitionsgleichungen

$s_i^{v+1} = \delta_i\left(S_k^v, E_g^v\right), mit\ s_i^{v+1} \in S_k^{v+1}$

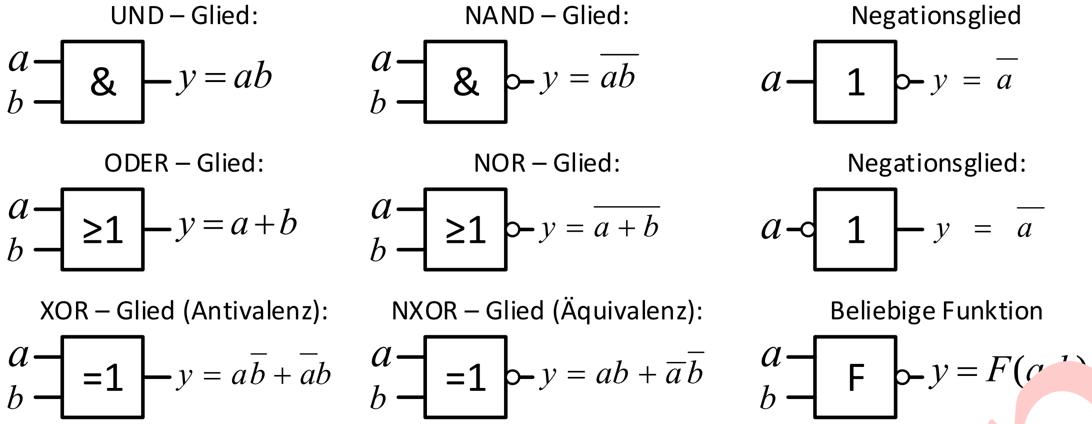
Ausgabefunktionen Medwedew $A_h^v = S_k^v$

Ausgabefunktionen Moore $A_h^v = \lambda_i\left(S_k^v\right)$

Ausgabefunktionen Mealy $A_h^v = \lambda_i\left(S_k^v, E_g^v\right)$

a, b, x, y, z : boolesche Variablen l : Literal f, g : boolesche Funktionen v_1 : p-Netz v_0 : n-Netz
s : Summe/Zustand c : Carry i : Eingang δ : Transitionsfunktion λ : Ausgabefunktion

Schaltsymbole

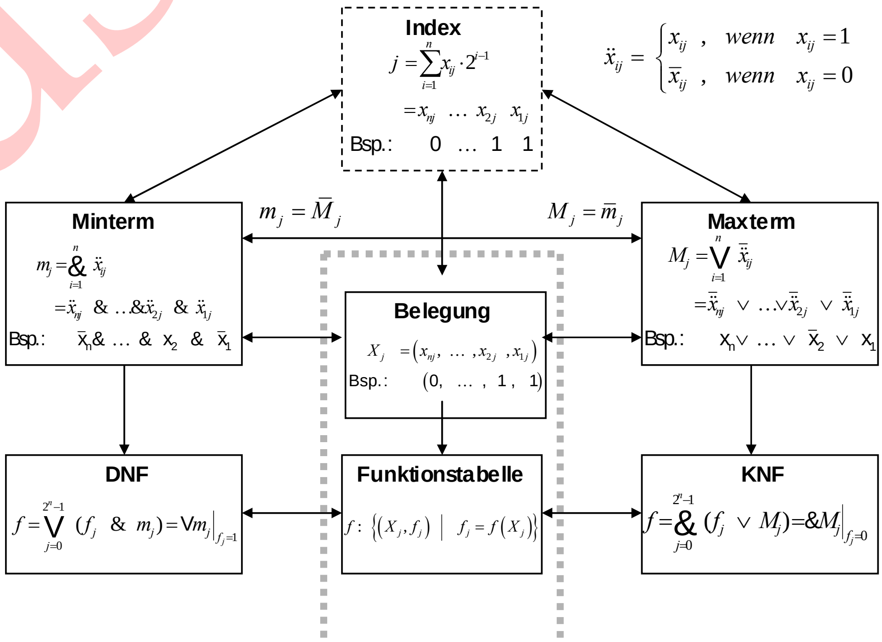


Zusammenstellung der wichtigsten Basissysteme

Zahl der Operatoren	Namen	Zeichen	Darstellung mit UND, ODER, NICHT	Darstellung von $\overline{a}$	$a\&b$	$a\vee b$
3	NICHT UND ODER	$y=\overline{a}$ $y=a\&b$ $y=a\vee b$	-	$\overline{a}$ - -	- $a\&b$ -	- - $a\vee b$
2	NICHT UND	$y=\overline{a}$ $y=a\&b$	-	$\overline{a}$ -	- $a\&b$	$a\vee b=\overline{\overline{a\vee b}}=\overline{\overline{a}\&\overline{b}}$
2	NICHT ODER	$y=\overline{a}$ $y=a\vee b$	-	$\overline{a}$ -	$a\&b=\overline{\overline{a\&b}}$ $=\overline{\overline{a}\vee\overline{b}}$	- $a\vee b$
2	UND ANTIVALENZ (Konstante 1)	$y=a\&b$ $y=a\oplus b$	$y=a\&b$ $y=\overline{a}\&b\vee a\&\overline{b}$	$\overline{a}=a\&1\vee\overline{a}\&\overline{1}$ $=a\oplus 1$	$a\&b$ -	$a\vee b$ $=a\oplus b\oplus(a\&b)$
1	NAND	$y=\overline{a\&b}$	$y=\overline{a\&b}$ $=\overline{a}\vee\overline{b}$	$\overline{a}=\overline{a\&a}=a\overline{\&}a$ oder $\overline{a}=\overline{a\&1}=a\overline{\&}1$	$a\&b=\overline{\overline{a\&b}}$ $=\overline{a\overline{\&}b}$ $=(a\overline{\&}b)\overline{\&}(a\overline{\&}b)$	$a\vee b=\overline{\overline{a}\&\overline{b}}$ $=\overline{a}\overline{\&}\overline{b}$ $=(a\overline{\&}a)\overline{\&}(b\overline{\&}b)$
1	NOR	$y=\overline{a\vee b}$	$y=\overline{a\vee b}$ $=\overline{a}\&\overline{b}$	$\overline{a}=\overline{a\vee a}=a\overline{\vee}a$ oder $\overline{a}=\overline{a\vee 0}=a\overline{\vee}0$	$a\&b=\overline{\overline{a}\vee\overline{b}}$ $=\overline{a}\overline{\vee}\overline{b}$ $=(a\overline{\vee}a)\overline{\vee}(b\overline{\vee}b)$	$a\vee b=\overline{\overline{a\vee b}}$ $=\overline{a\overline{\vee}b}$ $=(a\overline{\vee}b)\overline{\vee}(a\overline{\vee}b)$

Hauptsatz der Schaltalgebra

Beziehungen zwischen den Begriffen



a,b,x,y,z : boolesche Variablen l : Literal f,g : boolesche Funktionen v_1 : p-Netz v_0 : n-Netz
s : Summe/Zustand c : Carry i : Eingang δ : Transitionsfunktion λ : Ausgabefunktion