

Informatik – Übungsaufgaben

Digitallogik

Tobias Krähling
eMail: <Tobias.Kraehling@SemiByte.de>
Homepage: <www.SemiByte.de>

22.02.2007
Version: 1.5

Zusammenfassung

Die Übungsaufgaben stammen aus den Übungsaufgaben und Anwesenheitsaufgaben zur Vorlesung »Einführung in die Informatik« von Herrn Prof. Dr. Wassermann an der Ruhr-Universität Bochum im Wintersemester 2006/07.

In diesem Dokument sind sowohl die Übungsaufgaben wie auch Lösungsvorschläge enthalten.

Inhaltsverzeichnis

1. Übungsaufgaben	2
Frage 1 (Gatter-Schaltungen aus Feldeffekt-Transistoren)	2
Frage 2 (Logik-Gatter)	2
Frage 3 (Boole'sche Algebra)	2
Frage 4 (Boole'sche Funktionen)	2
Frage 5 (Disjunktive Normalform)	2
Frage 6 (Logische Schaltung)	3
Frage 7 (Flipflop-Schaltung)	3
Frage 8 (Flipflop-Schaltung)	3
Frage 9 (Multiplexer)	3
Frage 10 (Taktsignale)	3
Frage 11 (Mikrozähler)	4
Frage 12 (PLA)	4
Frage 13 (Addierer)	4
Frage 14 (Addierer)	4
Frage 15 (Carry-Select-Addierer)	4
Frage 16 (Carry-Look-Ahead-Addierer)	5
Frage 17 (Wallace Baum)	5
2. Lösungsvorschläge zu den Fragen	6

1. Übungsaufgaben

► Frage 1 (Gatter-Schaltungen aus Feldeffekt-Transistoren)

Man kann mit zwei in Serie geschalteten Transistoren ein NAND-Gatter und mit zwei in Reihe geschalteten ein NOR-Gatter implementieren, jeweils mit zwei Eingängen. Die beteiligten Transistoren sollen mit T_1 und T_2 bezeichnet werden und die drei Verbindungen G , S und D jedes Feldeffekt-Transistors sollen mit Index des Transistors versehen werden (z. B. bezeichnet G_2 das Gate von Transistor T_2).

Bei diesen Konstruktionen ist

- S_1 geerdet (mit Masse verbunden)
- S_2 geerdet oder mit D_1 verbunden
- D_1 mit S_2 oder dem Ausgang des Gatters verbunden und
- D_2 mit dem Ausgang des Gatters verbunden.

Angenommen, wir wollen ein Gatter mit drei Eingängen und somit mit drei Transistoren T_1 , T_2 und T_3 bauen, die nach der gleichen allgemeinen Regel miteinander verbunden werden dürfen:

- das Source jedes Transistors darf entweder mit Masse oder mit dem Drain eines oder mehrerer Transistoren mit *kleinerem* Index verbunden werden;
- das Drain jedes Transistors darf entweder mit dem Ausgang des Gatters oder mit dem Source eines oder mehrerer Transistoren mit *größerem* Index verbunden werden.

Zeichnen Sie alle Möglichkeiten von Schaltkreisen aus drei Transistoren, die nach diesem Prinzip und diesen Regeln gebaut sind, und geben Sie für jede Möglichkeit die boolesche Funktion von drei Variablen an, die das Gatter berechnet.

► Frage 2 (Logik-Gatter)

Der NOR-Baustein liefert genau dann eine 1 am Ausgang, wenn alle Eingangssignale 0 sind. Stellen Sie die folgenden boole'schen Ausdrücke nur mit Hilfe des NAND-Bausteins sowie nur mit Hilfe des NOR-Bausteins (jeweils mit zwei Eingängen) grafisch als Schaltung dar:

- \overline{ab}
- $\overline{a} + b$
- $a + b + c$

► Frage 3 (Boole'sche Algebra)

Beweisen oder widerlegen Sie folgende Gleichungen für $a, b, c \in \{0, 1\}$, indem Sie für beide Seiten der Gleichung eine Funktionswertetabelle (»Wahrheitstabelle«) aufstellen.

- $a + \overline{b}a = a$
- $\overline{a(\overline{a} + b)} = \overline{a} + \overline{b}$
- $(\overline{a} + b)(a + b + \overline{c}) = b + \overline{c}$
- $(a + b)(b + c)(a + \overline{c}) = ac + b\overline{c}$

► Frage 4 (Boole'sche Funktionen)

Geben Sie für alle 16 zweistelligen boole'schen Funktionen $f_i(a, b)$, $0 \leq i \leq 15$ einfache definierende Ausdrücke bei Verwendung der Bausteine NAND, NOR, NOT, AND und OR an.

► Frage 5 (Disjunktive Normalform)

- a) Überführen Sie den Ausdruck

$$\overline{\overline{(a + \bar{b})} \left(\bar{c} + \overline{(\bar{a}b) + (b\bar{c} + (\bar{a}\bar{b}))} \right)}$$

unter Ausnutzung der Rechenregeln der Boole'schen Algebra schrittweise in eine kanonische disjunktive Normalform.

- b) Wenden Sie das Verfahren von Quine-McCluskey (mit dem anschließenden weiteren Optimierungsschritt) auf folgenden boole'schen Ausdruck an:

$$abc + ab\bar{c} + a\bar{b}c + a\bar{b}\bar{c} + \bar{a}bc + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c}$$

► **Frage 6 (Logische Schaltung)**

In einem internationalen Gremium wollen die Länder A, B, C und D über eine gemeinsame Handelspolitik beraten. Beschlüsse in diesem Gremium werden angenommen, falls mindestens drei der Länder zustimmen oder falls die beiden größten Länder A und B zustimmen.

- a) Geben Sie eine Schaltfunktion »Zustimmung« für die Ermittlung des Abstimmungsergebnisses bei Eingaben A, B, C und D als Wahrheitstafel und in möglichst einfacher disjunktiver Normalform an.
- b) Zeichnen Sie die entsprechende Schaltung unter ausschließlicher Verwendung von AND-, OR- und NOT-Gattern.

► **Frage 7 (Flipflop-Schaltung)**

Erstellen Sie eine Flipflop-Schaltung aus NAND-Gattern und geben Sie zu jedem möglichen Eingang (R, S) und Ursprungszustand (Q, \bar{Q}) den Folgezustand $Q = \text{NAND}(R, \bar{Q})$ und $\bar{Q} = \text{NAND}(S, Q)$ an. Welches Eingangssignal (R, S) bewahrt den Zustand? Welches Signal setzt welchen Zustand und welches Eingangssignal ist unzulässig?

► **Frage 8 (Flipflop-Schaltung)**

Eine Flipflop-Schaltung kann als ein 1-Bit Datenspeicher benutzt werden. Ergänzen Sie das Schaltdiagramm für die Flipflop-Schaltung so, daß nur eine einzige Datenleitung D , aber zwei Steuerleitungen W (Schreibsignal) und L (Lesesignal) die Flipflop-Schaltung mit der Außenwelt verbinden und so daß folgende Verhalten zustande kommt:

Wenn das W -Signal 1 ist, dann wird der Zustand von D im Flipflop gesetzt (so daß der Ausgang Q des Flipflops den gleichen Zustand annimmt wie D). Wenn das L -Signal 1 ist, dann wird der Zustand des Ausgangs Q des Flipflops auf die Datenleitung D geschrieben, ohne daß der Flipflop seinen Zustand ändert. (Das Uhrensinal bleibt unberücksichtigt und muß nicht mit gezeichnet werden.)

► **Frage 9 (Multiplexer)**

- a) Zeichnen Sie das Schaltdiagramm eines Multiplexers mit zwei Dateneingängen (vergessen Sie nicht, daß der Multiplexer auch eine Anzahl von Steuereingängen hat).
- b) Realisieren Sie einen Multiplexer für 8 Dateneingangssignale nur mit Hilfe von Multiplexern aus Teil a) mit je zwei Dateneingängen.

► **Frage 10 (Taktsignale)**

Ein Computer hat eine Uhr mit einer Taktfrequenz von 4 GHz und somit mit einer Taktzykluszeit von 250 ps. Zur Verfügung stehen NOT, AND und OR Gatter mit einer Signaldurchlaufzeit von 50 ps.

Entwickeln Sie Schaltungen, die Taktsignale produzieren, die in jedem Taktzyklus genau einen Impuls mit Signalwert 1 der angegebenen Breite zum angegebenen Zeitpunkt haben (außerhalb des Impulses soll das Signal den Wert 0 haben).

- a) Impulsbreite 50 ps, beginnend 100 ps nach dem Anfang des Taktzyklus;
- b) Impulsbreite 50 ps, beginnend genau am Anfang des Taktzyklus;

► **Frage 11 (Mikrozähler)**

Entwerfen Sie die Schaltfunktionen für einen *rückwärts zählenden* Mikrozähler mit zehn Zuständen.

► **Frage 12 (PLA)**

Man programmiere die Schaltfunktionen

$$F(x_1, x_2, x_3) = (x_1x_3 + \bar{x}_2, \bar{x}_1x_2\bar{x}_3, x_1x_2x_3 + x_1x_2\bar{x}_3)$$

und

$$G(y_1, y_2, y_3, y_4, y_5) = (y_1y_2y_3 + y_4y_5, \bar{y}_1y_2y_4 + \bar{y}_3y_4y_5, y_1y_2y_3 + y_3y_4 + y_3\bar{y}_5, y_1y_3y_5)$$

in einem *einzigem* genügend groß dimensionierten PLA.

► **Frage 13 (Addierer)**

Wir definieren den *Aufwand* einer Schaltung als die Gesamtanzahl der Eingänge in AND- und OR-Gatter. Die *Laufzeit* bewerten wir durch die Maximalzahl *hintereinander* geschalteter AND- und OR-Gatter in einem Signalweg.

Beachten Sie bitte folgende Punkte: Die Schaltungen in dieser Aufgabe dürfen nur aus NOT-, AND- und OR-Gattern aufgebaut werden. Bei der Berechnung des Aufwandes und der Laufzeit brauchen für die genichteten Eingängen keine NOT-Gatter berücksichtigt werden, da die Flip-Flops, aus denen die Eingänge kommen, die genichteten Werte zur Verfügung stellen.

- a) Berechnen Sie Aufwand und Laufzeit für einen Half-Adder (HA) und einen Full-Adder (FA), wenn die Schaltungen mit Hilfe der disjunktiven Normalform realisiert werden.
- b) Entwerfen Sie eine Schaltung für einen Half-Adder mit Aufwand 7. Welche Laufzeit ergibt sich?
- c) Entwerfen Sie eine Schaltung für einen Full-Adder mit Aufwand 16. Berechnen Sie die Laufzeit.

(*Hinweis:* Nutzen Sie die Gatter der Schaltung für das Carry-Bit, um das Summen-Bit einfacher zu ermitteln. Verwenden Sie dann zwei Half-Adder, um einen Full-Adder zu konstruieren.)

► **Frage 14 (Addierer)**

Es seien die Zahlen $x = 55$, $y = 172$ und $z = 76$ in 8-Bit-Registern

$$X := [x_7, \dots, x_0], \quad Y := [y_7, \dots, y_0] \quad \text{und} \quad Z := [z_7, \dots, z_0]$$

abgelegt. Berechnen Sie

- a) $x + y$ mit einem »billigen« Addierer,
- b) $x + z$ mit dem von-Neumann-Addierer,
- c) $y + z$ mit dem asynchronen Carry-Ripple Addierer.

Geben Sie den Inhalt der jeweils beteiligten Register nach jedem Zyklus des entsprechenden Mikroprogramms an.

▸ Frage 15 (Carry-Select-Addierer)

Entwickeln Sie die Schaltung für einen zweistufigen 4-Bit Carry-Select-Addierer. »Zweistufig« soll heißen, daß das Addierwerk für 4-Bit Zahlen in zwei Hälften zerlegt wird, die in einer weiteren Stufe noch einmal zerlegt werden; im Endresultat sind dann viele *Einbit*additionen parallel auszuführen.

Die Einbitaddierer können Sie jeweils durch einen Kasten mit der richtigen Anzahl von Eingängen und Ausgängen darstellen; die Schaltung im Innern dieser Kästen müssen Sie nicht zeichnen.

▸ Frage 16 (Carry-Look-Ahead-Addierer)

- a) Entwerfen Sie die vollständige Schaltung für einen Carry-Look-Ahead-Addierer (CLA), welcher als Eingabe zwei 3-Bit Zahlen a und b erhält und die Summe berechnet.
- b) Geben Sie den Aufwand und die Laufzeit für den Addierer an.

▸ Frage 17 (Wallace Baum)

Entwerfen Sie einen Wallace Baum für die Multiplikation zweier 5-Bit Zahlen. Berechnen Sie damit das Quadrat von 5 und geben Sie die Zwischensummen nach jeder Addierstufe an.

2. Lösungsvorschläge zu den Fragen

► Frage 1 (Gatter-Schaltungen aus Feldeffekt-Transistoren)

Man kann mit zwei in Serie geschalteten Transistoren ein NAND-Gatter und mit zwei in Reihe geschalteten ein NOR-Gatter implementieren, jeweils mit zwei Eingängen. Die beteiligten Transistoren sollen mit T_1 und T_2 bezeichnet werden und die drei Verbindungen G , S und D jedes Feldeffekt-Transistors sollen mit Index des Transistors versehen werden (z. B. bezeichnet G_2 das Gate von Transistor T_2).

Bei diesen Konstruktionen ist

- S_1 geerdet (mit Masse verbunden)
- S_2 geerdet oder mit D_1 verbunden
- D_1 mit S_2 oder dem Ausgang des Gatters verbunden und
- D_2 mit dem Ausgang des Gatters verbunden.

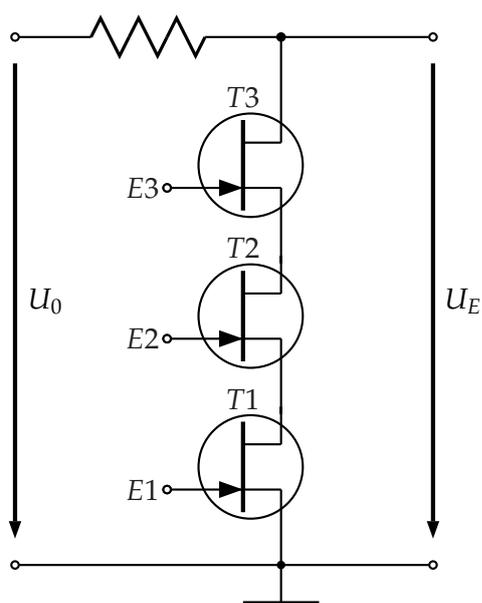
Angenommen, wir wollen ein Gatter mit drei Eingängen und somit mit drei Transistoren T_1 , T_2 und T_3 bauen, die nach der gleichen allgemeinen Regel miteinander verbunden werden dürfen:

- das Source jedes Transistors darf entweder mit Masser oder mit dem Drain eines oder mehrerer Transistoren mit *kleinerem* Index verbunden werden;
- das Drain jedes Transistors darf entweder mit dem Ausgang des Gatters oder mit dem Source eines oder mehrerer Transistoren mit *größerem* Index verbunden werden.

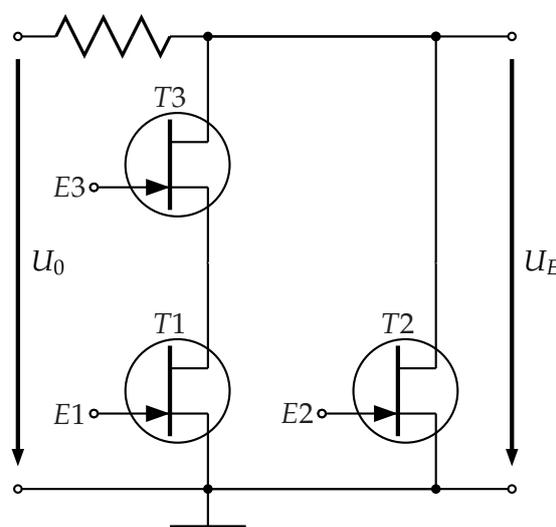
Zeichnen Sie alle Möglichkeiten von Schaltkreisen aus drei Transistoren, die nach diesem Prinzip und diesen Regeln gebaut sind, und geben Sie für jede Möglichkeit die boolesche Funktion von drei Variablen an, die das Gatter berechnet.

► Antwort 1

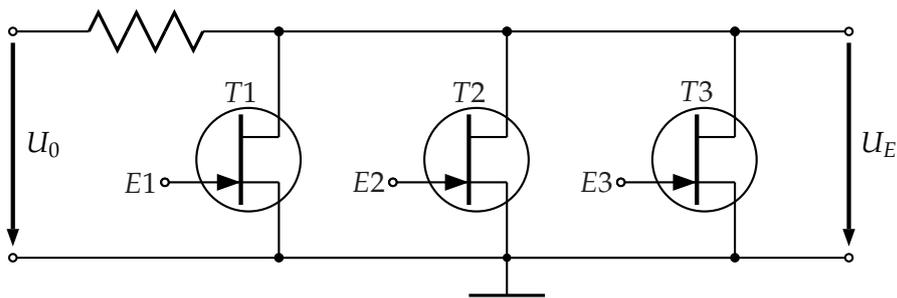
Es gibt 7 Schaltmöglichkeiten mit den angegebenen Einschränkungen:



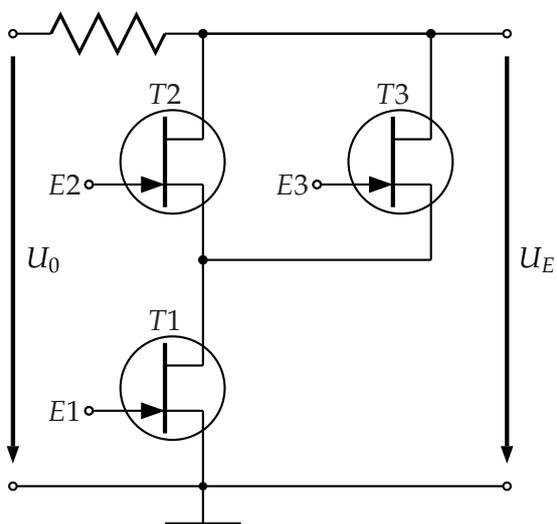
$$f(e_1, e_2, e_3) = \overline{e_1 e_2 e_3}$$



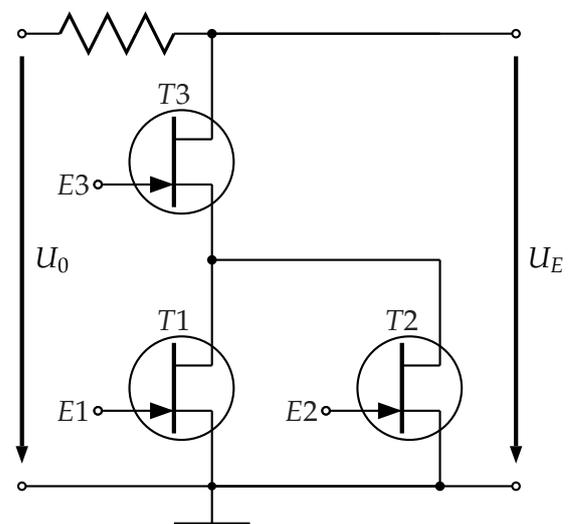
$$f(e_1, e_2, e_3) = \overline{e_1 e_3} + \overline{e_2}$$



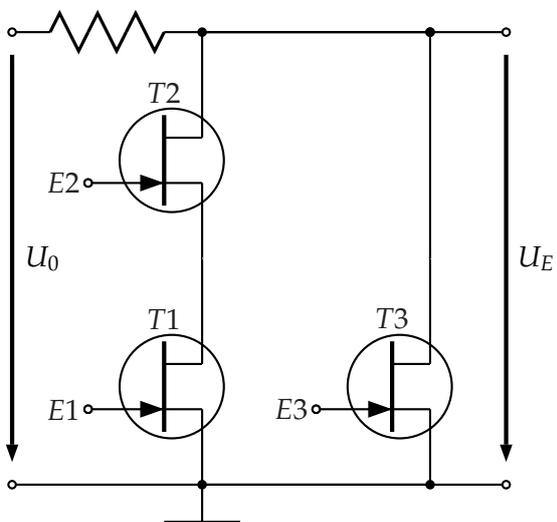
$$f(e_1, e_2, e_3) = \overline{e_1 + e_2 + e_3}$$



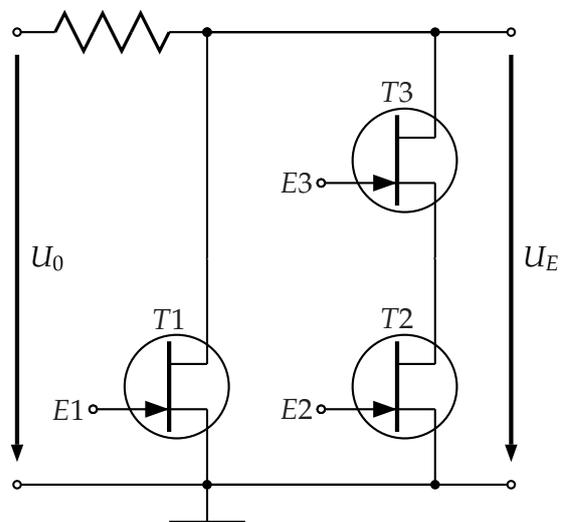
$$f(e_1, e_2, e_3) = \overline{e_1(e_2 + e_3)}$$



$$f(e_1, e_2, e_3) = \overline{(e_1 + e_2)e_3}$$



$$f(e_1, e_2, e_3) = \overline{e_1e_2 + e_3}$$



$$f(e_1, e_2, e_3) = \overline{e_1 + e_2e_3}$$

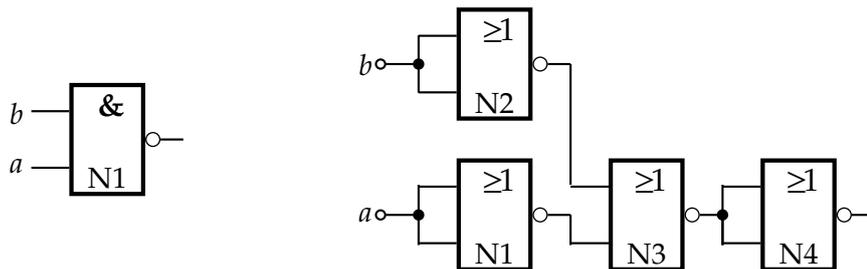
► **Frage 2 (Logik-Gatter)**

Der NOR-Baustein liefert genau dann eine 1 am Ausgang, wenn alle Eingangssignale 0 sind. Stellen Sie die folgenden boole'schen Ausdrücke nur mit Hilfe des NAND-Bausteins sowie nur mit Hilfe des NOR-Bausteins (jeweils mit zwei Eingängen) grafisch als Schaltung dar:

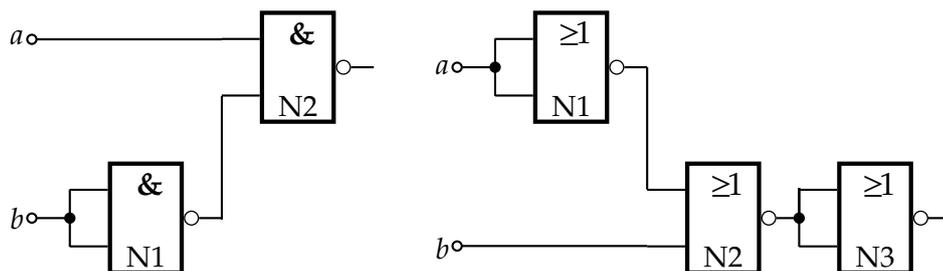
- a) \overline{ab}
- b) $\overline{a} + b$
- c) $a + b + c$

► **Antwort 2**

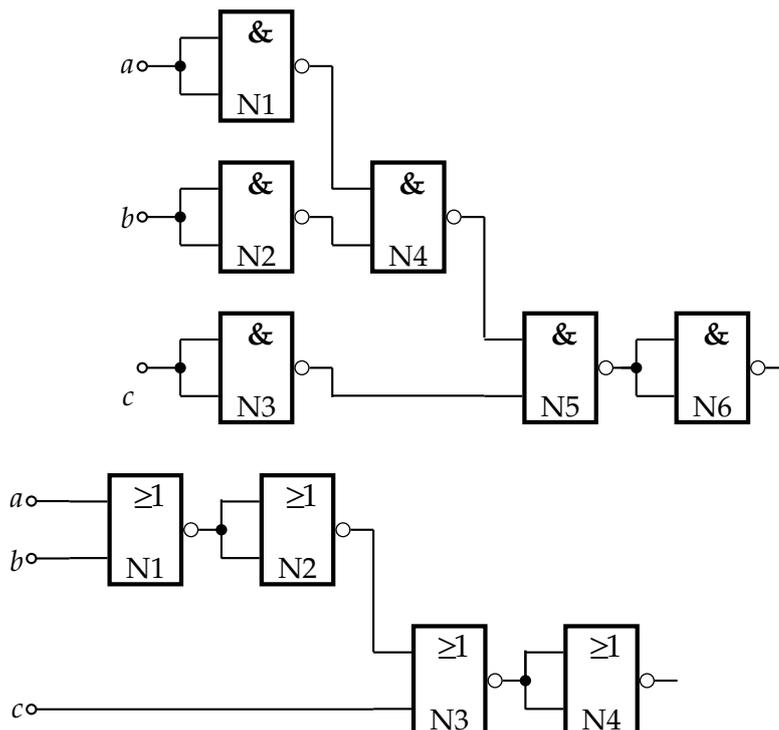
- a) \overline{ab}



- b) $\overline{a} + b$



- c) $a + b + c$



► **Frage 3 (Boole'sche Algebra)**

Beweisen oder widerlegen Sie folgende Gleichungen für $a, b, c \in \{0, 1\}$, indem Sie für beide Seiten der Gleichung eine Funktionswertetabelle (»Wahrheitstabelle«) aufstellen.

a) $a + \bar{b}a = a$

b) $\overline{a(\bar{a} + b)} = \bar{a} + \bar{b}$

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

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

► **Antwort 3**

a) $a + \bar{b}a = a$

a	b	\bar{b}	$\bar{b}a$	$a + \bar{b}a$
0	0	1	0	0
0	1	0	0	0
1	0	1	1	1
1	1	0	0	1

d. h. die Gleichung ist richtig;

b) $\overline{a(\bar{a} + b)} = \bar{a} + \bar{b}$

a	b	\bar{a}	\bar{b}	$\bar{a} + b$	$a(\bar{a} + b)$	$\overline{a(\bar{a} + b)}$	$\bar{a} + \bar{b}$
0	0	1	1	1	0	1	1
0	1	1	0	1	0	1	1
1	0	0	1	0	0	1	1
1	1	0	0	1	1	0	0

d. h. die Gleichung ist richtig;

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

a	b	c	\bar{a}	\bar{c}	$(\bar{a} + b)$	$(a + b + \bar{c})$	$(\bar{a} + b)(a + b + \bar{c})$	$b + \bar{c}$
0	0	0	1	1	1	1	1	1
0	0	1	1	0	1	0	0	0
0	1	0	1	1	1	1	1	1
0	1	1	1	0	1	1	1	1
1	0	0	0	1	0	1	0	1
1	0	1	0	0	0	1	0	0
1	1	0	0	1	1	1	1	1
1	1	1	0	0	1	1	1	1

d. h. die Gleichung ist falsch;

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

a	b	c	\bar{c}	$a + b$	$b + c$	$a + \bar{c}$	$(a + b)(b + c)(a + \bar{c})$	ac	$b\bar{c}$	$ac + b\bar{c}$
0	0	0	1	0	0	1	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0
0	1	0	1	1	1	1	1	0	1	1
0	1	1	0	1	1	0	0	0	0	0
1	0	0	1	1	0	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	0	1	1	1	1	1	0	1

d. h. die Gleichung ist richtig;

► **Frage 4 (Boole'sche Funktionen)**

Geben Sie für alle 16 zweistelligen boole'schen Funktionen $f_i(a, b)$, $0 \leq i \leq 15$ einfache definierende Ausdrücke bei Verwendung der Bausteine NAND, NOR, NOT, AND und OR an.

► **Antwort 4**

Die 16 zweistelligen boolschen Funktionen sind:

a	0	0	1	1	
b	0	1	0	1	
f_0	0	0	0	0	$(ab)(\overline{ab})$ oder $(ab)\overline{a}$ oder $a\overline{a}$
f_1	0	0	0	1	ab
f_2	0	0	1	0	\overline{ab}
f_3	0	1	0	0	$\overline{a}b$
f_4	1	0	0	0	$a + b$
f_5	0	0	1	1	$a + ab$ oder a
f_6	0	1	0	1	$b + ab$ oder b
f_7	1	0	0	1	$ab + \overline{a}b$
f_8	0	1	1	0	$\overline{ab + \overline{a}b}$ oder $\overline{a}b + \overline{a}b$
f_9	1	0	1	0	$\overline{a + b + \overline{a}b}$ oder \overline{b}
f_{10}	1	1	0	0	$\overline{a} + ab$ oder \overline{a}
f_{11}	0	1	1	1	$a + b$
f_{12}	1	1	0	1	$\overline{a} + b$
f_{13}	1	0	1	1	$a + \overline{b}$
f_{14}	1	1	1	0	\overline{ab}
f_{15}	1	1	1	1	$\overline{ab} + a$ oder $a + \overline{a}$

Die Ausdrücke, um die boolschen Funktionen aus den bekannten Bausteinen zusammenzusetzen sind in der Kurzschreibweise angegeben. Bei einigen sind, falls möglich, die boolschen Funktionen auch nur als Funktion einer Variablen angegeben, wenn die zweite irrelevant wäre und ignoriert werden könnte.

► **Frage 5 (Disjunktive Normalform)**

a) Überführen Sie den Ausdruck

$$\overline{\overline{(a + \bar{b})} \left(\bar{c} + \overline{(\bar{a}b) + (b\bar{c} + (\bar{a}\bar{b}))} \right)}$$

unter Ausnutzung der Rechenregeln der Boole'schen Algebra schrittweise in eine kanonische disjunktive Normalform.

b) Wenden Sie das Verfahren von Quine-McCluskey (mit dem anschließenden weiteren Optimierungsschritt) auf folgenden boole'schen Ausdruck an:

$$abc + ab\bar{c} + a\bar{b}c + a\bar{b}\bar{c} + \bar{a}bc + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c}$$

► **Antwort 5**

a)

$$\begin{aligned} & \overline{\overline{(a + \bar{b})} \left(\bar{c} + \overline{(\bar{a}b) + (b\bar{c} + (\bar{a}\bar{b}))} \right)} \\ \Leftrightarrow & \overline{(a + \bar{b}) + \overline{\bar{c} + (\bar{a}b) + (b\bar{c} + (\bar{a}\bar{b}))}} \\ \Leftrightarrow & (a + \bar{b}) + \bar{c} \overline{(\bar{a}b) + (b\bar{c} + \bar{a}\bar{b})} \\ \Leftrightarrow & a + \bar{b} + c \overline{(\bar{a}b) + b\bar{c} + \bar{a}\bar{b}} \\ \Leftrightarrow & a + \bar{b} + c (\bar{a} + \bar{b} + b\bar{c} + \bar{a}\bar{b}) \\ \Leftrightarrow & a + \bar{b} + \bar{a}c + \bar{b}c + b\bar{c}c + \bar{a}\bar{b}c \\ \Leftrightarrow & a + \bar{b} + \bar{a}c + \bar{b}c + b + \bar{a}\bar{b}c \\ \Leftrightarrow & ab + \bar{a}\bar{b} + \bar{a}\bar{b} + \bar{a}\bar{b}c + \bar{a}\bar{b}c + \bar{a}\bar{b}c + \bar{a}\bar{b}c + \bar{a}\bar{b}c \\ \Leftrightarrow & ab + \bar{a}\bar{b} + \bar{a}\bar{b} + \bar{a}\bar{b}c + \bar{a}\bar{b}c + \bar{a}\bar{b}c \\ \Leftrightarrow & abc + ab\bar{c} + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c \\ \Leftrightarrow & abc + ab\bar{c} + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c \end{aligned}$$

b)

$$\begin{array}{cccccccc} abc & + & ab\bar{c} & + & a\bar{b}c & + & a\bar{b}\bar{c} & + & \bar{a}bc & + & \bar{a}\bar{b}c & + & \bar{a}\bar{b}\bar{c} \\ \text{BÄ} & 111 & + & 110 & + & 101 & + & 100 & + & 011 & + & 001 & + & 000 \end{array}$$

Gruppe	BÄ		BÄ		BÄ	
0	000	√	00-	√	-0-	
			-00	√		
1	001	√	0-1	√	--1	
	100	√	-01	√	1--	
			10-	√		⇒ a + \bar{b} + c
			1-0	√		
2	011	√	-11	√		
	101	√	1-1	√		
	110	√	11-	√		
3	111	√				

► **Frage 6 (Logische Schaltung)**

In einem internationalen Gremium wollen die Länder A, B, C und D über eine gemeinsame Handelspolitik beraten. Beschlüsse in diesem Gremium werden angenommen, falls mindestens drei der Länder zustimmen oder falls die beiden größten Länder A und B zustimmen.

- a) Geben Sie eine Schaltfunktion »Zustimmung« für die Ermittlung des Abstimmungsergebnisses bei Eingaben A, B, C und D als Wahrheitstafel und in möglichst einfacher disjunktiver Normalform an.
- b) Zeichnen Sie die entsprechende Schaltung unter ausschließlicher Verwendung von AND-, OR- und NOT-Gattern.

► **Antwort 6**

a)

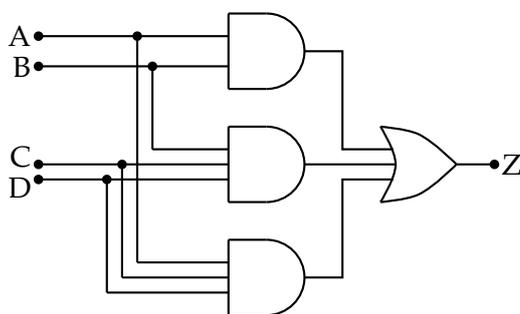
a	b	c	d	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$$ab + bcd + acd + abd + abc + abcd$$

$$\Leftrightarrow ab + bcd + acd$$

Die letzten drei Terme können wegfallen, da die Aussage bereits wahr ist, wenn ab wahr ist, unabhängig von c und d .
 Auf das selbe Ergebnis kommt man über das Verfahren von Quine-McCluskey.

b)

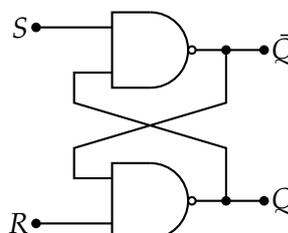


► **Frage 7 (Flipflop-Schaltung)**

Erstellen Sie eine Flipflop-Schaltung aus NAND-Gattern und geben Sie zu jedem möglichen Eingang (R,S) und Ursprungszustand (Q,\bar{Q}) den Folgezustand $Q = \text{NAND}(R,\bar{Q})$ und $\bar{Q} = \text{NAND}(S,Q)$ an. Welches Eingangssignal (R,S) bewahrt den Zustand? Welches Signal setzt welchen Zustand und welches Eingangssignal ist unzulässig?

► **Antwort 7**

S	R	$Q\bar{Q}$	$\text{NAND}(R,\bar{Q})$	$\text{NAND}(S,Q)$
0	0	01	1	1
0	1	01	0	1
1	0	01	1	1
1	1	01	0	1
0	0	10	1	1
0	1	10	1	1
1	0	10	1	0
1	1	10	1	0



allg:	S	R	Q	\bar{Q}
	0	0	1	1
	0	1	0	1
	1	0	1	0
	1	1	bleibt wie es war	

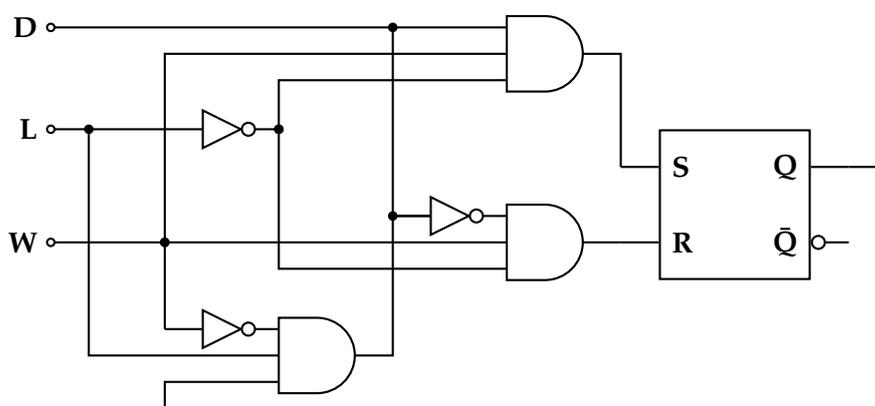
Das Eingangssignal $S = 1$, $R = 1$ bewahrt den Zustand, unzulässig ist die Kombination $S = 0, R = 0$.
Wenn $S \neq R$, so ist $Q = S$

► **Frage 8 (Flipflop-Schaltung)**

Eine Flipflop-Schaltung kann als ein 1-Bit Datenspeicher benutzt werden. Ergänzen Sie das Schaltdiagramm für die Flipflop-Schaltung so, daß nur eine einzige Datenleitung D , aber zwei Steuerleitungen W (Schreibsignal) und L (Lesesignal) die Flipflop-Schaltung mit der Außenwelt verbinden und so daß folgende Verhalten zustande kommt:

Wenn das W -Signal 1 ist, dann wird der Zustand von D im Flipflop gesetzt (so daß der Ausgang Q des Flipflops den gleichen Zustand annimmt wie D). Wenn das L -Signal 1 ist, dann wird der Zustand des Ausgangs Q des Flipflops auf die Datenleitung D geschrieben, ohne daß der Flipflop seinen Zustand ändert. (Das Uhrensinal bleibt unberücksichtigt und muß nicht mit gezeichnet werden.)

► **Antwort 8**

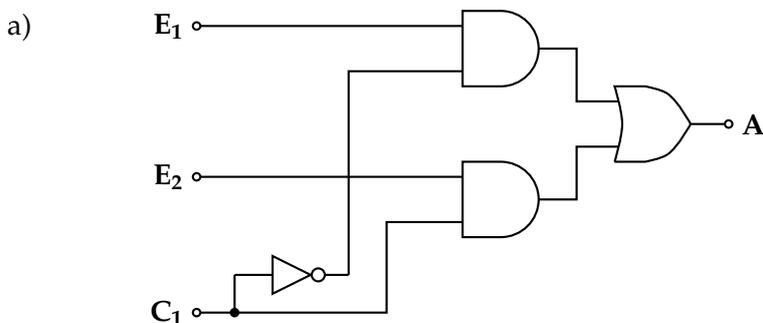


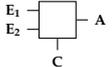
Die Schaltung berücksichtigt auch, daß L und W nicht gleichzeitig 1 sein dürfen, ist dieser Zustand vorhanden, so ändert sich der Zustand des SR-FL nicht.

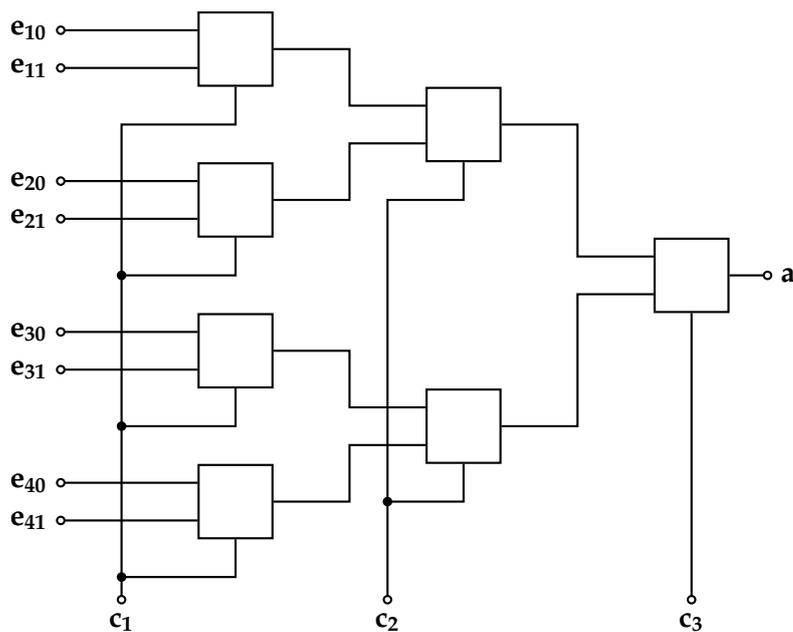
► Frage 9 (Multiplexer)

- a) Zeichnen Sie das Schaltdiagramm eines Multiplexers mit zwei *Dateneingängen* (vergessen Sie nicht, daß der Multiplexer auch eine Anzahl von *Steuereingängen* hat).
- b) Realisieren Sie einen Multiplexer für 8 Dateneingangssignale nur mit Hilfe von Multiplexern aus Teil a) mit je zwei Dateneingängen.

► Antwort 9



b) der Multiplexer aus a) wird im folgenden als  dargestellt.



Funktionstabelle, welche Dateneingänge mit welchen Steuereingängen ausgewählt werden:

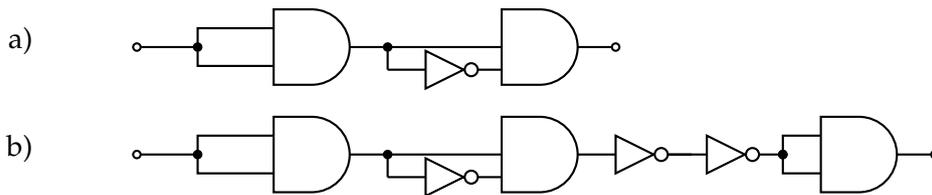
c ₃	c ₂	c ₁	a
0	0	0	e ₁₀
0	0	1	e ₁₁
0	1	0	e ₂₀
0	1	1	e ₂₁
1	0	0	e ₃₀
1	0	1	e ₃₁
1	1	0	e ₄₀
1	1	1	e ₄₁

► Frage 10 (Taktsignale)

Ein Computer hat eine Uhr mit einer Taktfrequenz von 4 GHz und somit mit einer Taktzykluszeit von 250 ps. Zur Verfügung stehen NOT, AND und OR Gatter mit einer Signaldurchlaufzeit von 50 ps.

Entwickeln Sie Schaltungen, die Taktsignale produzieren, die in jedem Taktzyklus genau einen Impuls mit Signalwert 1 der angegebenen Breite zum angegebenen Zeitpunkt haben (außerhalb des Impulses soll das Signal den Wert 0 haben).

- Impulsbreite 50 ps, beginnend 100 ps nach dem Anfang des Taktzyklus;
- Impulsbreite 50 ps, beginnend genau am Anfang des Taktzyklus;

► Antwort 10

► **Frage 11 (Mikrozähler)**

Entwerfen Sie die Schaltfunktionen für einen *rückwärts zählenden* Mikrozähler mit zehn Zuständen.

► **Antwort 11**

$x_1x_2x_3x_4$	$z_1z_2z_3z_4$
1001	1000
1000	0111
0111	0110
0110	0101
0101	0100
0100	0011
0011	0010
0010	0001
0001	0000
0000	1001

$$z_1 = x_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$$

$$z_2 = x_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3x_4 + \bar{x}_1x_2x_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3x_4$$

$$z_3 = x_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3x_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3x_4$$

$$z_4 = x_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$$

Optimierung

z_1 kann nicht weiter optimiert werden.

$$z_2 \quad x_1\bar{x}_2\bar{x}_3\bar{x}_4 \quad \bar{x}_1x_2x_3x_4 \quad \bar{x}_1x_2x_3\bar{x}_4 \quad \bar{x}_1x_2\bar{x}_3x_4$$

$$\quad 1000 \quad 0111 \quad 0110 \quad 0101$$

Gr. 1 1000

Gr. 2 0110√ 011- ⇒ $z_2 = x_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3 + \bar{x}_1x_2x_4$
 0101√ 01 - 1

Gr. 3 0111√

$$z_3 \quad x_1\bar{x}_2\bar{x}_3\bar{x}_4 \quad \bar{x}_1x_2x_3x_4 \quad \bar{x}_1x_2\bar{x}_3\bar{x}_4 \quad \bar{x}_1\bar{x}_2x_3x_4$$

$$\quad 1000 \quad 0111 \quad 0100 \quad 0011$$

Gr. 1 1000

0100

Gr. 2 0011√ 0 - 11 ⇒ $z_3 = x_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_3x_4$

Gr. 3 0111√

$$z_4 \quad x_1\bar{x}_2\bar{x}_3\bar{x}_4 \quad \bar{x}_1x_2x_3\bar{x}_4 \quad \bar{x}_1x_2\bar{x}_3\bar{x}_4 \quad \bar{x}_1\bar{x}_2x_3\bar{x}_4 \quad \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$$

$$\quad 1000 \quad 0110 \quad 0100 \quad 0010 \quad 0000$$

Gr. 0 0000√ -000

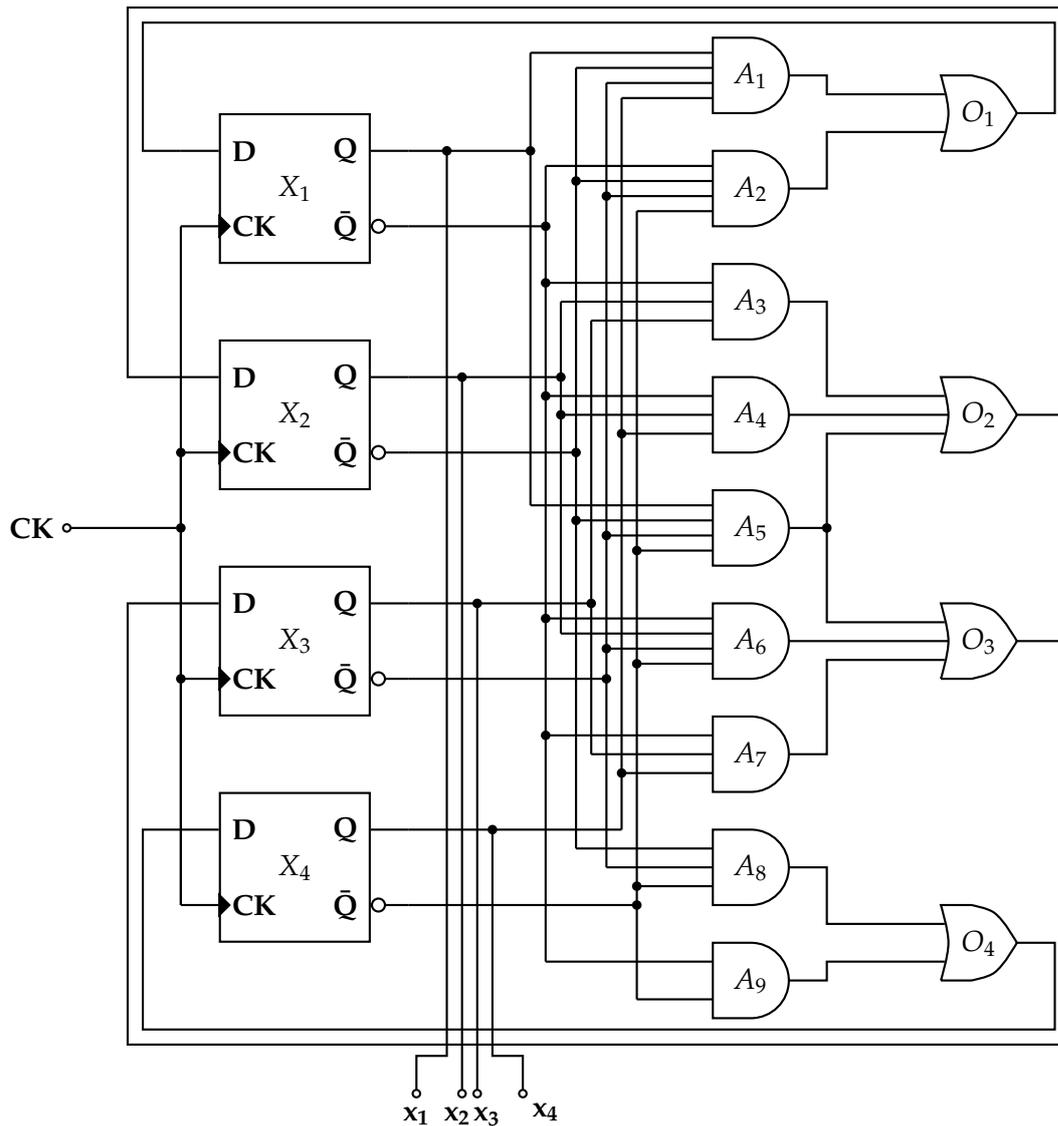
Gr. 1 1000√ 0 - 00√ 0 - -0

0100√ 00 - 0√

⇒ $z_4 = \bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_4$

0010√ 01 - 0√

Gr. 2 0110√ 0 - 10√



$$A1 = x_1 \bar{x}_2 \bar{x}_3 x_4$$

$$A2 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$$

$$A3 = \bar{x}_1 x_2 x_3$$

$$A4 = \bar{x}_1 x_2 x_4$$

$$A5 = x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$$

$$A6 = \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4$$

$$A7 = \bar{x}_1 x_3 x_4$$

$$A8 = \bar{x}_2 \bar{x}_3 \bar{x}_4$$

$$A9 = \bar{x}_1 \bar{x}_4$$

(doppelt verwendet)

$$O1 = A1 + A2$$

$$O2 = A3 + A4 + A5$$

$$O3 = A5 + A6 + A7$$

$$O4 = A8 + A9$$

► **Frage 12 (PLA)**

Man programmiere die Schaltfunktionen

$$F(x_1, x_2, x_3) = (x_1x_3 + \bar{x}_2, \bar{x}_1x_2\bar{x}_3, x_1x_2x_3 + x_1x_2\bar{x}_3)$$

und

$$G(y_1, y_2, y_3, y_4, y_5) = (y_1y_2y_3 + y_4y_5, \bar{y}_1y_2y_4 + \bar{y}_3y_4y_5, y_1y_2y_3 + y_3y_4 + y_3\bar{y}_5, y_1y_3y_5)$$

in einem *einzigem* genügend groß dimensionierten PLA.

► **Antwort 12**

$$\left. \begin{aligned} f_1 &= x_1x_3 + \bar{x}_2 \\ f_2 &= \bar{x}_1x_2\bar{x}_3 \\ f_3 &= x_1x_2x_3 + x_1x_2\bar{x}_3 \end{aligned} \right\} \text{mit } \begin{aligned} x_1 &= e_1 \\ x_2 &= e_2 \\ x_3 &= e_3 \end{aligned}$$

$$\left. \begin{aligned} g_1 &= y_1y_2y_3 + y_4y_5 \\ g_2 &= \bar{y}_1y_2y_4 + \bar{y}_3y_4y_5 \\ g_3 &= y_1y_2y_3 + y_3y_4 + y_3\bar{y}_5 \\ g_4 &= y_1y_3y_5 \end{aligned} \right\} \text{mit } \begin{aligned} y_1 &= e_4 \\ y_2 &= e_5 \\ y_3 &= e_6 \\ y_4 &= e_7 \\ y_5 &= e_8 \end{aligned}$$

	1	1	1	1	1	1	1	1	1	1	1		
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
$e_1 \rightarrow$	2	0	0	2	2	0	0	0	0	0	0	0	
$\bar{e}_1 \rightarrow$	0	0	2	0	0	0	0	0	0	0	0	0	
$e_2 \rightarrow$	0	0	2	2	2	0	0	0	0	0	0	0	
$\bar{e}_2 \rightarrow$	0	2	0	0	0	0	0	0	0	0	0	0	
$e_3 \rightarrow$	2	0	0	2	0	0	0	0	0	0	0	0	
$\bar{e}_3 \rightarrow$	0	0	2	0	2	0	0	0	0	0	0	0	
$e_4 \rightarrow$	0	0	0	0	0	2	0	0	0	0	0	2	
$\bar{e}_4 \rightarrow$	0	0	0	0	0	0	0	2	0	0	0	0	
$e_5 \rightarrow$	0	0	0	0	0	2	0	2	0	0	0	0	
$\bar{e}_5 \rightarrow$	0	0	0	0	0	0	0	0	0	0	0	0	
$e_6 \rightarrow$	0	0	0	0	0	2	0	0	0	2	2	2	
$\bar{e}_6 \rightarrow$	0	0	0	0	0	0	0	2	0	0	0	0	
$e_7 \rightarrow$	0	0	0	0	0	0	2	2	2	2	0	0	
$\bar{e}_7 \rightarrow$	0	0	0	0	0	0	0	0	0	0	0	0	
$e_8 \rightarrow$	0	0	0	0	0	0	2	0	2	0	0	2	
$\bar{e}_8 \rightarrow$	0	0	0	0	0	0	0	0	0	0	2	0	
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
$0 \rightarrow$	1	1	0	0	0	0	0	0	0	0	0	0	$\rightarrow f_1$
$0 \rightarrow$	0	0	1	0	0	0	0	0	0	0	0	0	$\rightarrow f_2$
$0 \rightarrow$	0	0	0	1	1	0	0	0	0	0	0	0	$\rightarrow f_3$
$0 \rightarrow$	0	0	0	0	0	1	1	0	0	0	0	0	$\rightarrow g_1$
$0 \rightarrow$	0	0	0	0	0	0	0	1	1	0	0	0	$\rightarrow g_2$
$0 \rightarrow$	0	0	0	0	0	1	0	0	0	1	1	0	$\rightarrow g_3$
$0 \rightarrow$	0	0	0	0	0	0	0	0	0	0	0	1	$\rightarrow g_4$

► **Frage 13 (Addierer)**

Wir definieren den *Aufwand* einer Schaltung als die Gesamtanzahl der Eingänge in AND- und OR-Gatter. Die *Laufzeit* bewerten wir durch die Maximalzahl *hintereinander* geschalteter AND- und OR-Gatter in einem Signalweg.

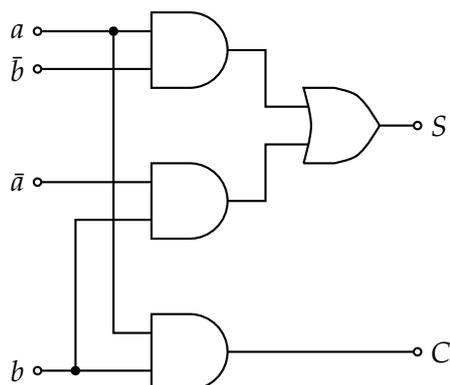
Beachten Sie bitte folgende Punkte: Die Schaltungen in dieser Aufgabe dürfen nur aus NOT-, AND- und OR-Gattern aufgebaut werden. Bei der Berechnung des Aufwandes und der Laufzeit brauchen für die genichteten Eingängen keine NOT-Gatter berücksichtigt werden, da die Flip-Flops, aus denen die Eingänge kommen, die genichteten Werte zur Verfügung stellen.

- Berechnen Sie Aufwand und Laufzeit für einen Half-Adder (HA) und einen Full-Adder (FA), wenn die Schaltungen mit Hilfe der disjunktiven Normalform realisiert werden.
- Entwerfen Sie eine Schaltung für einen Half-Adder mit Aufwand 7. Welche Laufzeit ergibt sich?
- Entwerfen Sie eine Schaltung für einen Full-Adder mit Aufwand 16. Berechnen Sie die Laufzeit.

(*Hinweis:* Nutzen Sie die Gatter der Schaltung für das Carry-Bit, um das Summen-Bit einfacher zu ermitteln. Verwenden Sie dann zwei Half-Adder, um einen Full-Adder zu konstruieren.)

► **Antwort 13**

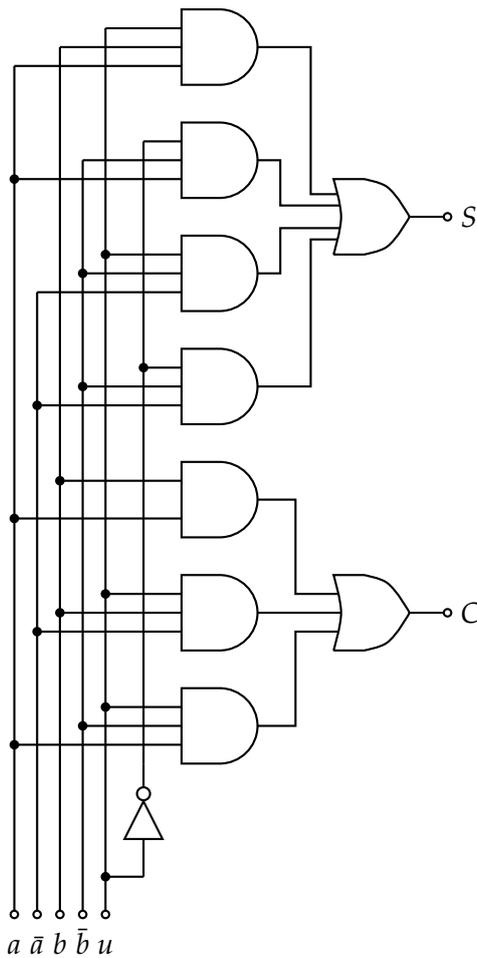
- a) • Half-Adder



$$\text{Für } S = a\bar{b} + \bar{a}b \\ C = ab$$

$$\begin{aligned} \text{Aufwand für } S &= 6 \\ C &= 2 \\ \text{Gesamtaufwand} &= 8 \\ \text{Laufzeit} &= 2 \end{aligned}$$

• Full-Adder

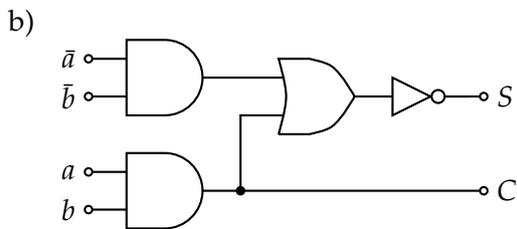


$$\begin{aligned}
 S' &= \bar{a}\bar{b} + \bar{a}b \\
 S &= S'\bar{u} + \bar{S}'u \\
 &= (\bar{a}\bar{b} + \bar{a}b)\bar{u} + \overline{(\bar{a}\bar{b} + \bar{a}b)}u \\
 &= \bar{a}\bar{b}\bar{u} + \bar{a}b\bar{u} + (\bar{a}\bar{b}\bar{a}b)u \\
 &= \bar{a}\bar{b}\bar{u} + \bar{a}b\bar{u} + (\bar{a} + b)(a + \bar{b})u \\
 &= \bar{a}\bar{b}\bar{u} + \bar{a}b\bar{u} + \bar{a}b\bar{u} + abu \\
 C' &= ab \\
 C &= C' + (\bar{a}\bar{b} + \bar{a}b)u \\
 &= ab + (\bar{a}\bar{b} + \bar{a}b)u \\
 &= ab + \bar{a}b\bar{u} + \bar{a}bu
 \end{aligned}$$

Aufwand = 27 (28)

Laufzeit = 2 (3)

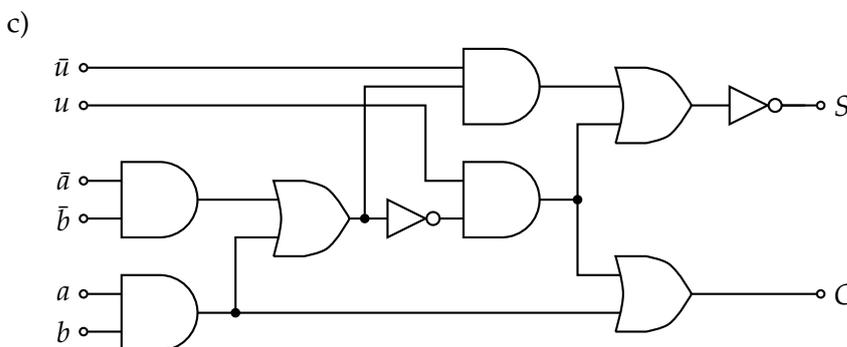
Die Werte in Klammern geben jeweils die Werte an, wenn für u das negierte Signal nicht automatisch zur Verfügung steht.



$$\bar{a}\bar{b} + \bar{a}b \Leftrightarrow \overline{ab + \bar{a}\bar{b}}$$

Aufwand = 7

Laufzeit = 3



Aufwand 16

Laufzeit 6

Steht \bar{u} nicht zur Verfügung, erhöht sich der Aufwand um 1 (für das NOT-Gatter), die Laufzeit bleibt gleich.

► **Frage 14 (Addierer)**

Es seien die Zahlen $x = 55$, $y = 172$ und $z = 76$ in 8-Bit-Registern

$$X := [x_7, \dots, x_0], \quad Y := [y_7, \dots, y_0] \quad \text{und} \quad Z := [z_7, \dots, z_0]$$

abgelegt. Berechnen Sie

- $x + y$ mit einem »billigen« Addierer,
- $x + z$ mit dem von-Neumann-Addierer,
- $y + z$ mit dem asynchronen Carry-Ripple Addierer.

Geben Sie den Inhalt der jeweils beteiligten Register nach jedem Zyklus des entsprechenden Mikroprogramms an.

► **Antwort 14**

$$\begin{array}{ll} x = 55 & \Rightarrow X := 00110111 \\ y = 172 & \Rightarrow Y := 10101100 \\ z = 76 & \Rightarrow Z := 01001100 \end{array}$$

a)

X = 00110111	z = 8
Y = 10101100	Ca = 0
X = 10011011	z = 7
Y = 01010110	Ca = 0
X = 11001101	z = 6
Y = 00101011	Ca = 0
X = 01100110	z = 5
Y = 00010101	Ca = 1
X = 00110011	z = 4
Y = 00001010	Ca = 1
X = 00011001	z = 3
Y = 00000101	Ca = 1
X = 10001100	z = 2
Y = 00000010	Ca = 1
X = 11000110	z = 1
Y = 00000001	Ca = 0
X = 01100011	z = 0
Y = 00000000	Ca = 0

Berechnung beendet $\Rightarrow x + y = 11100011$

b)

X = 00110111	O = 0
Z = 01001100	
X = 01111011	O = 0
Z = 00001000	
X = 01110011	O = 0
Z = 00010000	
X = 01100011	O = 0
Z = 00100000	
X = 01000011	O = 0
Z = 01000000	
X = 00000011	O = 0
Z = 10000000	
X = 10000011	O = 0
Z = 00000000	

Berechnung beendet, $\sum_i z_i = 0 \Rightarrow x + z = 10000011$

c)

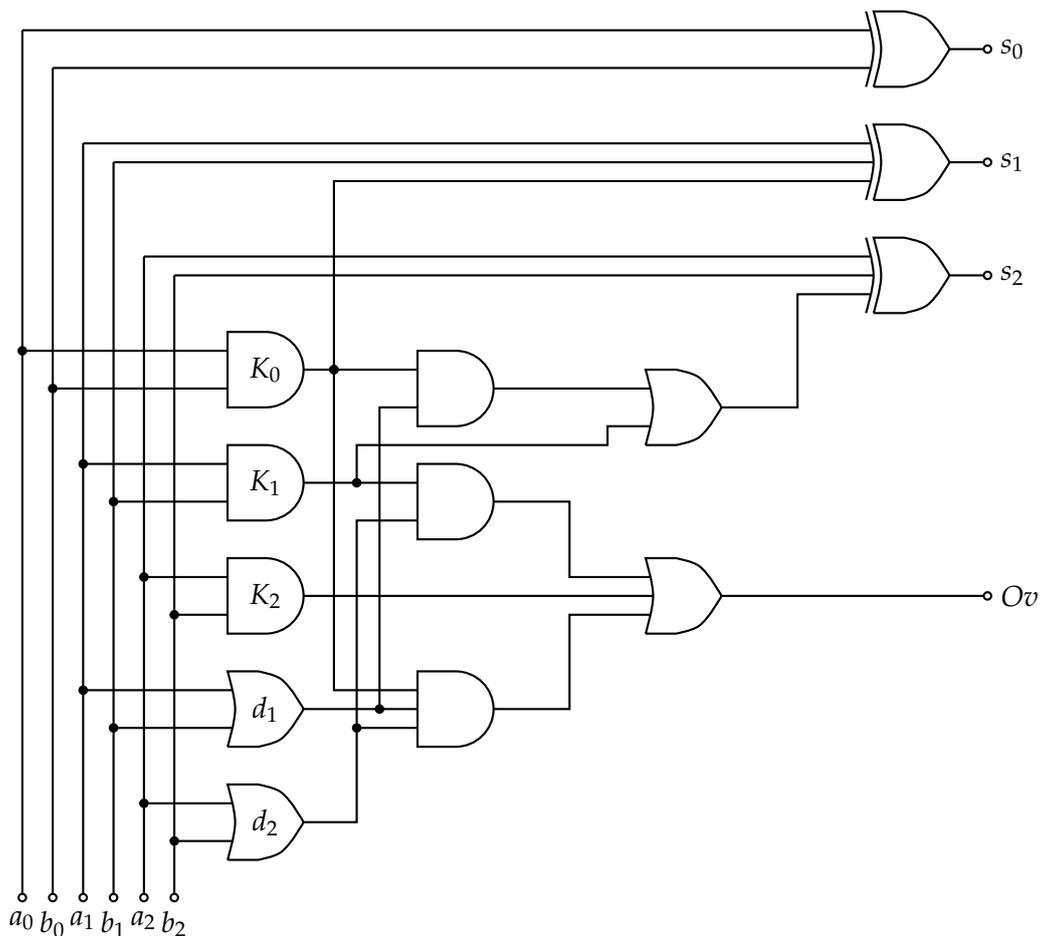
Schritt 0	f: 00000000		
	h: 00000001		
1	f: 00001100		
	h: 00010011	Y	10101100
2	f: 00001100	Z	01001100
	h: 00110011	$\Rightarrow C_{i-1}$	00011000
3	f: 00001100	y + z	11111000
	h: 01110011		
4	f: 00001100		
	h: 11110011		

► **Frage 16 (Carry-Look-Ahead-Addierer)**

- a) Entwerfen Sie die vollständige Schaltung für einen Carry-Look-Ahead-Addierer (CLA), welcher als Eingabe zwei 3-Bit Zahlen a und b erhält und die Summe berechnet.
 b) Geben Sie den Aufwand und die Laufzeit für den Addierer an.

► **Antwort 16**

- a)
- $$c_0 = K_0$$
- $$c_1 = K_1 + d_1 c_0 = K_1 + d_1 K_0$$
- $$c_2 = K_2 + d_2 c_1 = K_2 + d_2 K_1 + d_1 d_2 K_0$$
- $$s_0 = a_0 \oplus b_0$$
- $$s_1 = a_1 \oplus b_1 \oplus c_0$$
- $$s_2 = a_2 \oplus b_2 \oplus c_1$$
- $$Ov = c_2$$
- $$K_n = a_n b_n$$
- $$d_n = a_n + b_n$$

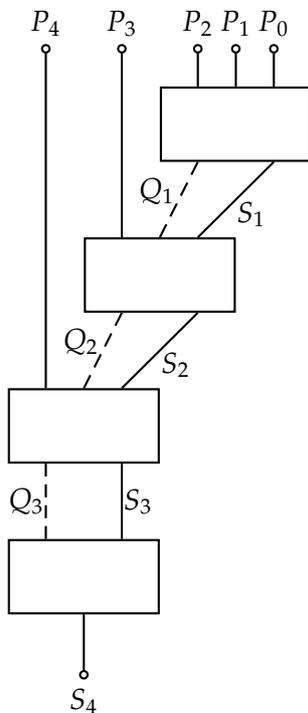


- b)
- Aufwand: 17
 - Laufzeit: 4

► **Frage 17 (Wallace Baum)**

Entwerfen Sie einen Wallace Baum für die Multiplikation zweier 5-Bit Zahlen. Berechnen Sie damit das Quadrat von 5 und geben Sie die Zwischensummen nach jeder Addierstufe an.

► **Antwort 17**



- $5_{10} = 101_2$
- $P_0 = 00101$
- $P_1 = 0$
- $P_2 = 10100$
- $P_3 = 0$
- $P_4 = 0$
- $S_1 = 10001$
- $Q_1 = 01000$
- $S_2 = 11001$
- $Q_2 = 0$
- $S_3 = 11001$
- $Q_3 = 0$
- $S_4 = 11001$

Liste der Versionen

Version	Datum	Bearbeiter	Bemerkung
1.0	04.02.2007	Krä	Dokumenterstellung
1.1	05.02.2007	Krä	Frage 8 hinzugefügt
1.2	09.02.2007	Krä	Frage 9–12 hinzugefügt
1.3	13.02.2007	Krä	Frage 13–14 hinzugefügt
1.4	17.02.2007	Krä	Frage 15–16 hinzugefügt
1.5	22.02.2007	Krä	Frage 16 (Carry-Look-Ahead-Addierer) hinzugefügt