100 Übungsaufgaben zu Grundlagen der Informatik

Band II: Technische Informatik

von

Lukas König
Karlsruher Institut für Technologie (KIT)

Friederike Pfeiffer-Bohnen
Karlsruher Institut für Technologie (KIT)

Prof. Dr. Hartmut Schmeck
Karlsruher Institut für Technologie (KIT)

Oldenbourg Verlag München

Lektorat: Johannes Breimeier
Herstellung: Tina Bonertz
Grafik: Irina Apetrei
Einbandgestaltung: hauser lacour

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Library of Congress Cataloging-in-Publication Data
A CIP catalog record for this book has been applied for at the Library of Congress.

Rosenheimer Straße 143, 81671 München, Deutschland
www.degruyter.com/oldenbourg
Ein Unternehmen von De Gruyter

Gedruckt in Deutschland

Dieses Papier ist alterungsbeständig nach DIN/ISO 9706.

ISBN 978-3-486-75696-8
eISBN 978-3-486-78130-4

Inhaltsverzeichnis

Vorwort

Diese Aufgabensammlung in zwei Bänden über die Grundlagen der Informatik ist als Übungsbuch und Nachschlagewerk für Studierende der Informatik und verwandter Studiengänge konzipiert. Sie kann zur Prüfungsvorbereitung, vorlesungsbegleitend oder zur Wissensauffrischung genutzt werden. Der vorliegende zweite Band behandelt die Grundlagen der Technischen Informatik. Dazu gehören die Themen Rechnerarchitektur, Speicherorganisation, CMOS, Schaltnetze, Schaltwerke, Dateiorganisation, Betriebssysteme, Kodierung, Zahlendarstellung, Fehlererkennung und -korrektur, Verschlüsselung, Programmierung und Internettechnologie.

Entstehung

Die Aufgabensammlung ist in den Jahren 2009 bis 2013 im Rahmen der Vorlesung „Grundlagen der Informatik II“ entstanden, die als Grundlagenvorlesung für die Studiengänge Wirtschaftsingenieurwesen, Technische Volkswirtschaftslehre und Wirtschaftsmathematik am Karlsruher Institut für Technologie (KIT) üblicherweise für das dritte Semester angeboten wird. Dabei wurde dem Wunsch der Studierenden nach einer größeren Aufgabenfülle zur Prüfungsvorbereitung und nach Aufgaben, die den Stoff in verständlicher und intuitiver Weise veranschaulichen, nachgegangen. Es ist jedoch schwierig, gerade Anfängern eine intuitive Veranschaulichung der sehr formalen Inhalte zu ermöglichen, die erst über mehrere Ecken ihren praktischen Nutzen entfalten. Oft entsteht eine Kluft zwischen den Dozenten, die eine korrekte, in das Gesamtbild passende Sicht vermitteln wollen, und den Studierenden, denen dieses Gesamtbild zu Beginn noch nicht verständlich ist und für die eine schrittweise Heranführung, möglicherweise über Umwege, hilfreicher wäre. Um diesem Problem zu begegnen, wurden die Ideen für die Mehrzahl der Aufgaben und deren Lösungen nicht von Dozenten geliefert, sondern von Tutoren, also von studentischen Hilfskräften, die zur Unterstützung des Übungsbetriebs eingesetzt werden. Diese sind hauptsächlich Studierende, welche selbst die Vorlesung im Vorsemester gehört haben. So entstanden Aufgaben, die sich aus Studierendensicht mit den essentiellen Verständnisproblemen der jeweiligen Inhalte beschäftigen und auch in Studierendensprache die Lösung dieser Probleme darstellen. Die Aufgaben wurden teilweise in den vergangenen zwei Vorlesungszyklen den Studierenden zur Prüfungsvorbereitung angeboten und stießen dabei auf sehr gute Resonanz. Natürlich entziehen sich die Autoren dennoch nicht der Verantwortung, ausgewogene, hochwertige und fehlerfreie Aufgaben in diesem Buch zur Verfügung zu stellen. Die Aufgaben, die jetzt den Hauptteil dieser Aufgabensammlung bilden, wurden aus einem noch größeren Pool studentischer Aufgaben ausgewählt, sie wurden einzeln mehrfach überarbeitet, korrigiert, ergänzt und in ein einheitliches Format gebracht.

Struktur

Jeder Band der Aufgabensammlung ist in zwei Hauptteile gegliedert, von denen der erste eine Zusammenfassung der theoretischen Grundlagen und die Aufgabenstellungen enthält und der zweite Lösungen zu den Aufgaben. Der erste Hauptteil ist inhaltlich nach Kapiteln gegliedert,

die jeweils mit einer Einführung in das Thema und einer Erläuterung aller für das Lösen der Aufgaben nötigen Voraussetzungen beginnen. In diesem Rahmen wurde auch versucht, auf typische Probleme und Missverständnisse einzugehen, die Studierende erfahrungsgemäß häufig mit den jeweiligen Inhalten haben. Im Anschluss an die Einführung folgt in jedem Kapitel der Aufgabenteil, der Aufgabenstellungen unterschiedlicher Schwierigkeitsgrade enthält. Die Schwierigkeit einer Aufgabe wird zu deren Beginn durch die Symbole

- $\star$ leicht,
- $\star^\star$ mittel oder
- $\star^{\star^\star}$ schwer

gekennzeichnet. Einige wenige Aufgaben sind durch das Symbol $\star^{\star^\star}$ als sehr schwer bezeichnet; die Lösung dieser Aufgaben kann unter Umständen sehr viel Zeit beanspruchen und sollte nicht unterschätzt werden. Zu jeder Aufgabe gibt es im zweiten Hauptteil einen Lösungsvorschlag mit ausführlichen Erklärungen und Rechenwegen. Allgemeine Grundlagen werden ganz am Ende jedes Bandes im Anhang behandelt.

Vorlesungsaufzeichnungen

Die Vorlesung „Grundlagen der Informatik II“, auf deren Inhalten die Aufgabensammlung beruht, wird seit Jahren aufgezeichnet, und die Videos sind für den öffentlichen Zugriff freigeschaltet. Im Rahmen der Arbeit an diesem Buch wurden die Videos neu zusammengeschnitten, sodass sie thematisch zu den Kapiteln passen. Am Ende der Einführung jedes Kapitels ist ein Link abgedruckt, unter dem man zu dem entsprechenden Thema das Vorlesungsvideo abrufen kann. Das Video kann sowohl im Browser angesehen als auch im AVI-Format heruntergeladen werden.

Diskussionsforum

Im Rahmen der Forschung im Bereich der Verbesserung der Lehre in universitären Großveranstaltungen wurde in den Vorlesungszyklen der Wintersemester 2011/12 und 2012/13 der Vorlesung „Grundlagen der Informatik II“ ein Konzept entwickelt, das die Diskussion von Lösungen und Problemen einzelner Aufgaben zwischen Studierenden und Dozenten ermöglicht [Pfe+12]. Im Zentrum steht ein Forum, das für jede Aufgabe einen Thread enthält, in dem die Aufgabe diskutiert werden kann. Dieses Forum wird für die Buchversion der Aufgabensammlung fortgeführt und kann unter der folgenden Adresse abgerufen werden:

`http://www.dasinfobuch.de/links/Forum.html`

Um eine bestimmte Aufgabe im Forum wiederzufinden, ist jeder Aufgabe eine fünfstellige ID vorangestellt, beispielsweise „END-AA“. Die Threads des Forums sind mit genau diesen IDs

betitelt, die Diskussion zur Aufgabe END-AA findet also im Thread END-AA des Forums statt. Im Forum können die in der Vergangenheit bereits in über 2000 Beiträgen diskutierten Themen eingesehen oder neue eigene Beiträge verfasst werden. Selbstverständlich werden auch zukünftig die Dozenten im Forum weiter mitdiskutieren und die Probleme und Lösungen kommentieren.

Zusammen mit Vorlesungsaufzeichnungen und Diskussionsforum stellt diese Aufgabensammlung ein interaktives Gesamtkonzept dar, mit dem die Autoren hoffen, der Leserin und dem Leser auf eine ansprechende und verständliche Art die Grundlagen der Informatik nahebringen zu können. In diesem Sinne wünschen sie ihnen eine angenehme Lektüre und viel Erfolg beim Lernen und beim Lösen der Aufgaben.

Danksagung

Die Autoren bedanken sich für die kreative Mitwirkung an den Aufgaben bei Jonas Benterbusch, Philippe Blättchen, Marcel Braith, Leonard Brauck, Johannes Burchardt, Jose Antonio Cepeda, Vivian Cheng, Jacob Eberhardt, Theresa Euring, Yvonne Freytag , Philip Grefe, Anna-Lena Hau, Christiane Haubitz, Matthias Hauser, Stefanie Häuser, Michael Heck, Jonas Hemlein, Patrick Henkes, Yannick Hilgers, Lisa Hoffmann, Julian Holz, Lukas Jansen, Maximilian Jentsch, Fabian Jost, Claus Kadelka, Tobias Käfer, Nico Kaltenbacher, Fabian Kerstholt, Christian Kiefer, Isabelle Krämer, Martin Kreuser, Daniel Kübler, Melanie Lampert, Tobias Lurz, Florian Mägerlein, Christina Marx, Bastian Prell, Jördis Rappl, Adam Rodacki, Benedict Schendzielorz, Sebastian Schienle, Irena Schips, Sven Schömer, Marcel Schrumpf, Anja Schuler, Alexander Sigmund, Birgit Spitz, Eva Steffes, Tobias Stengel, Michael Vormittag, Simon Wiedemann, Lea Wild und Kai Windscheid.

Für Arbeiten an den Vorlesungsaufzeichnungen und der Verlinkung des Forums danken sie Solveig Gronau und Caroline Strunz.

Für das Durchsehen der Texte hinsichtlich Grammatik und Stil danken sie Christian Gitte, Christian Hirsch, Sebastian Kochanneck, Ingo Mauser, Sabrina Merkel, Marc Mültin, Daniel Pathmaperuma, Fabian Rigoll, Felix Vogel und Micaela Wünsche.

Karlsruhe, September 2013

Lukas König, Friederike Pfeiffer-Bohnen, Hartmut Schmeck

1 Schaltnetze und Schaltwerke

Einführung

Schaltnetze und Schaltwerke werden genutzt, um Boolesche Abbildungen technisch umzusetzen. Als logische Bausteine finden sie Verwendung in Prozessoren, Speicherplatinen und anderen Bauteilen moderner Computer, Smart-Phones usw. Dafür werden auf CMOS-Technologie (vgl. Kapitel 2) basierende Realisierungen von Elementarfunktionen für logische Verknüpfungen wie AND, OR, NOT, NAND, NOR, XOR usw. zusammengefügt, um komplexere Funktionalitäten zu erfüllen. Das können beispielsweise Berechnungen wie Addition, Subtraktion, Multiplikation und Division auf Dualzahlen sein, logische Operationen wie das Verschieben von Werten im Speicher oder die Speicherung einzelner Bits in sogenannten Flipflops.

Schaltalgebra

Der Arbeitsweise von Schaltnetzen und Schaltwerken liegt eine Logik zugrunde, die durch die sogenannte Schaltalgebra definiert wird. Um ihren Zweck zur Beschreibung des Verhaltens elektronischer Bauteile zu erfüllen, die aus Elementen wie Schaltern, Relais, Transistoren usw. bestehen, muss die Schaltalgebra das Verhalten und Zusammenspiel dieser Elemente formalisieren. Die Elemente der Schaltalgebra werden mit den Schaltzuständen bistabiler Schaltelemente, beispielsweise Schalter, Relais, Dioden und Transistoren, identifiziert. **Bistabile Schaltelemente** weisen zwei stabile Schaltzustände auf, die je nach Kontext als offen/geschlossen, nichtleitend/leitend, hohes/niedriges Potential usw. bezeichnet werden. Diesen beiden Zuständen werden die sogenannten **Schaltwerte** (oder **Schaltkonstanten**) 0 ($false$) und 1 ($true$) zugeordnet. Daher kann die Menge M, auf der die Schaltalgebra beruht, mit der Booleschen Menge identifiziert werden: $M = \mathbb{B}$. Die Verknüpfungen der Schaltalgebra werden bezeichnet als

- $\wedge$ (Boolesches Produkt, **Konjunktion**; realisiert durch Reihenschaltung von Schaltelementen),
- $\vee$ (Boolesche Summe, **Disjunktion**; realisiert durch Parallelschaltung von Schaltelementen),
- $\bar{}$ (Boolesches **Komplement** oder Negation; realisiert durch Ruhekontaktschaltung).

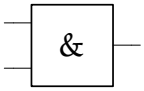
Man überzeugt sich leicht, dass diese Operationen den Axiomen der Booleschen Algebra genügen; die so definierte Schaltalgebra $(\mathbb{B}; \wedge, \vee, \bar{})$ ist also eine Boolesche Algebra (vgl. Anhang A.5).

Wir bezeichnen als (binäre) **Schaltvariable** eine Variable, die genau zwei Werte annehmen kann und somit mit der Menge $\mathbb{B}$ identifizierbar ist. Eine Funktion $f : \mathbb{B}^n \to \mathbb{B}$ bezeichnen wir als n-stellige **Schaltfunktion** und deren technische Realisierung als **Schaltung** mit n **Eingängen** und einem **Ausgang**.

Schaltungen zur Realisierung spezieller (ausgezeichneter) schaltalgebraischer Verknüpfungen nennt man **Schaltgatter** oder **Gatter**. Gatter werden als Rechtecke dargestellt, die eine Beschriftung zur Identifikation tragen und (meist) links einen waagerechten Strich pro Eingang und (meist) rechts einen waagerechten Strich für den Ausgang haben. Zu den Gattern zählen wir insbesondere die **elementaren Gatter**:

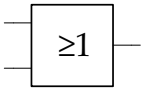
- **AND-Gatter** (Konjunktionsglied): Realisierung von $f : \mathbb{B}^2 \to \mathbb{B} : (a, b) \mapsto a \wedge b$;

  Darstellung:



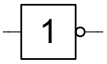
- **OR-Gatter** (Disjunktionsglied): Realisierung von $f : \mathbb{B}^2 \to \mathbb{B} : (a, b) \mapsto a \vee b$;

  Darstellung:



- **NOT-Gatter** (Negationsglied): Realisierung von $f : \mathbb{B} \to \mathbb{B} : a \mapsto \overline{a}$;

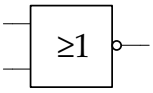
  Darstellung:



Weitere gebräuchliche Gatter sind:

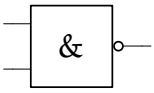
- **NOR-Gatter** (Peirce-Funktion): Realisierung von $f : \mathbb{B}^2 \to \mathbb{B} : (a, b) \mapsto \overline{a \vee b}$;

  Darstellung:



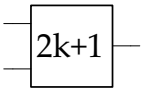
- **NAND-Gatter** (Sheffer-Funktion): Realisierung von $f : \mathbb{B}^2 \to \mathbb{B} : (a, b) \mapsto \overline{a \wedge b}$;

  Darstellung:



- **XOR-Gatter** (Antivalenz/Paritätsfunktion): Realisierung von $f : \mathbb{B}^2 \to \mathbb{B} : (a, b) \mapsto (a \wedge \overline{b}) \vee (\overline{a} \wedge b)$;

  Darstellung:



Für die zweistelligen unter diesen Gattern können, unter Verallgemeinerung der zugehörigen Schaltfunktion, auch mehr als zwei Eingänge realisiert werden, da die zugrunde liegenden Funktionen alle assoziativ sind.

Die Schaltalgebra erlaubt weiterhin die Verknüpfung mehrerer Gatter zu einer komplexeren Schaltstruktur, wobei auf Darstellungsebene einfach die Ausgänge von Gattern mit den Eingängen anderer Gatter verbunden werden. Daraus ergeben sich Schaltnetze oder Schaltwerke, die beliebig viele Eingänge und Ausgänge und eine beliebig komplexe Verknüpfungslogik haben können. Solange es dabei keine Rückkopplungen gibt, reden wir von Schaltnetzen, sonst von Schaltwerken. Im Folgenden werden Schaltnetze und Schaltwerke formal definiert.

Schaltnetze

Ein Schaltnetz F ist die technische Realisierung einer Abbildung

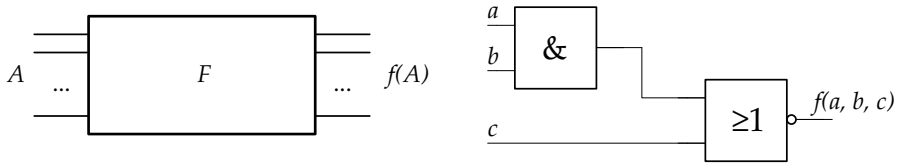
$$f : \mathbb{B}^n \to \mathbb{B}^m : (a_1, a_2, \ldots, a_n) \mapsto f(A) = (f_1(A), f_2(A), \ldots, f_m(A))$$

mit

- $A =_{def} (a_1, a_2, \ldots, a_n)$;
- $a_1, a_2, \ldots, a_n$: Eingaben (Werte an den Eingängen) des Schaltnetzes F;
- $f_1, f_2, \ldots, f_m$: n-stellige Schaltfunktionen, die die Ausgaben (Werte an den Ausgängen) $f_1(A), f_2(A), \ldots, f_m(A)$ des Schaltnetzes definieren.

Die Funktion f ist also eine Zusammenfassung von m jeweils n-stelligen Schaltfunktionen, und das Schaltnetz F ist eine Kombination aus m Schaltungen. Das Schaltverhalten von F ist dabei „kombinatorisch“, d. h. die Ausgaben hängen nur von den Eingaben ab (vgl. Schaltwerke). Gatter sind spezielle Schaltnetze mit $m = 1$ und beliebig vielen Eingaben.

Als Verallgemeinerung von Gattern stellen wir auch Schaltwerke als Rechtecke mit Eingängen und Ausgängen dar, oder wir zeichnen die inneren Verbindungen zwischen den an dem Schaltwerk beteiligten Gattern.

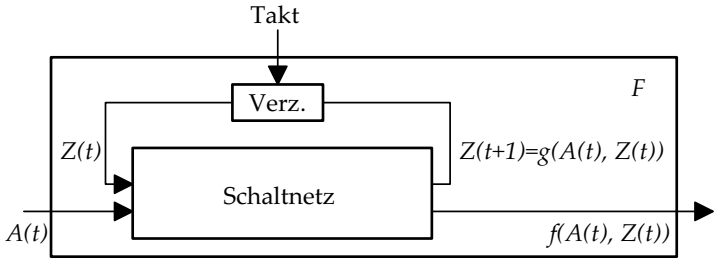


Die von einem Schaltnetz berechneten Funktionen können beispielsweise durch eine **Wertetabelle** angegeben werden, die für jeden der n Eingänge und m Ausgänge eine Spalte und insgesamt 2^n Zeilen enthält. Außerdem kann man für jede Boolesche Funktion einen **Booleschen Ausdruck** angeben (vgl. Anhang A.5) oder ein **Binary Decision Diagram (BDD)** berechnen (vgl. Kapitel 3).

(Synchrone) Schaltwerke

Schaltwerke stellen eine Verallgemeinerung von Schaltnetzen dar, indem die Ausgaben hier

nicht nur von den aktuellen Eingaben abhängen, sondern zusätzlich von endlich vielen vorausgegangenen Eingaben. Dafür ist ein „Gedächtnis“ notwendig, das bei Schaltwerken in Form sogenannter **innerer Zustände** vorhanden ist (in dieser Hinsicht hängen Schaltwerke eng mit endlichen Automaten zusammen, vgl. Band 1, Kapitel 2). Auf Darstellungsebene kann man sich die Realisierung dieser inneren Zustände durch die Rückführung von Ausgängen eines Schaltnetzes zu Eingängen seiner Teilschaltungen vorstellen.



Formal ist ein Schaltwerk F die technische Realisierung zweier Abbildungen:

$$f : \mathbb{B}^n \times \mathbb{B}^r \to \mathbb{B}^m : (A, Z) \mapsto f(A, Z) = (f_1(A, Z), f_2(A, Z), \ldots, f_m(A, Z))$$

mit $A =_{def} (a_1, a_2, \ldots, a_n)$ und $Z =_{def} (z_1, z_2, \ldots, z_r)$ sowie

$$g : \mathbb{B}^n \times \mathbb{B}^r \to \mathbb{B}^r : (A, Z) \mapsto g(A, Z) = (g_1(A, Z), g_2(A, Z), \ldots, g_r(A, Z))$$

mit

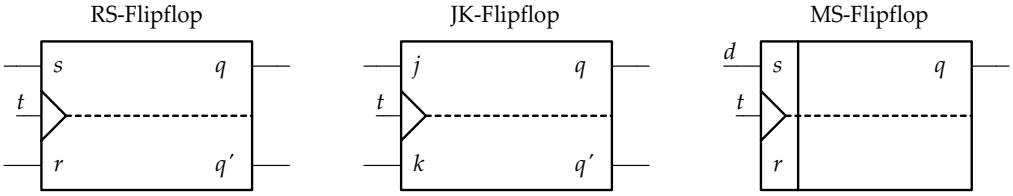
- $a_1, a_2, \ldots, a_n$: Eingaben des Schaltwerkes F;
- $z_1, z_2, \ldots, z_r$: innere Zustände des Schaltwerkes F;
- $f_1, f_2, \ldots, f_m$: Schaltfunktionen, die die Ausgaben $f_1(A, Z), f_2(A, Z), \ldots, f_m(A, Z)$ des Schaltwerkes definieren;
- $g_1, g_2, \ldots, g_r$: Schaltfunktionen, die den Übergang zu den neuen inneren Zuständen $g_1(A, Z), g_2(A, Z), \ldots, g_r(A, Z)$ des Schaltwerkes definieren.

Der Zustand eines Schaltwerkes heißt stabil, falls $g(A, Z) = Z$ und sonst instabil. Zur Rückführung der den inneren Zustand darstellenden Schaltnetzausgänge werden **Verzögerungsglieder** genutzt, die eine Verzögerung der Übertragung eines Signals um eine gewisse Zeit τ ermöglichen. Wir denken uns diese Glieder allerdings implizit an den rückführenden Leitungen und verzichten oft auf eine explizite Darstellung.

Um eine klar definierte sequentielle Abfolge von Zuständen und Ausgaben eines Schaltwerkes zu erhalten, nutzen wir ein Taktsignal als zusätzliche Eingabe (daher „synchrone“ Schaltwerke; bei asynchronen Schaltwerken ist es zunächst unklar, wann man ein Ergebnis „abrufen“ kann und wann es sich stattdessen um einen instabilen Zwischenzustand handelt). Das Taktsignal (oder einfach **Takt**) wechselt periodisch zwischen den Werten 0 und 1, wobei eine Periode die konstante Zeit τ dauert. Nun betrachten wir nur noch diskrete Zeitpunkte $t \cdot \tau$ mit $t \in \mathbb{N}_0$ und

nennen diese **Taktzeitpunkte** (vereinfachend spricht man meist von Taktzeitpunkten t anstatt von $t \cdot \tau$). Die Zeitspanne τ heißt **Taktzeit** und wird folgendermaßen bestimmt: Die Zeit, bis das Schaltnetz aus einer Eingabe A den Ausgabevektor $f(A)$ erzeugt hat, heißt **Schaltzeit** τ_s; beträgt τ_s höchstens τ_{max} Zeiteinheiten, so wählt man $\tau \geq \tau_{max}$. Zustandsänderungen erfolgen nun nur noch während der Taktzeiten, und erst beim Takt „erscheint“ das neue $Z(t) = g(A(t-1), Z(t-1))$ am Ausgang des Verzögerungsgliedes. Der Takt wirkt also wie eine „Schleuse“ für das zu übertragende Signal. Der innere Anfangszustand $Z(0)$ des Schaltwerkes kann unbekannt sein oder fest mit Anfangswerten initialisiert.

Schaltwerke können insbesondere zur zeitlichen Speicherung von Signalen genutzt werden, beispielsweise bei der Realisierung von Registern. Zu diesem Zweck werden sogenannte **Flipflops** genutzt, die jeweils genau einen Wert (0 oder 1) für eine gewisse Dauer speichern. Wir betrachten dabei (synchrone) RS-, JK- und MS-Flipflops und nutzen für diese die folgenden Schaltzeichen:



Alle synchronen Flipflops haben einen Takteingang t. Bei **RS-Flipflops** bewirkt der Takteingang, dass in den Speicher nur geschrieben werden kann, wenn das Taktsignal eine 1 liefert. Die zusätzlichen beiden Eingänge s und r werden zum Einstellen des gespeicherten Werts benutzt. Wird s („set“) auf 1 gesetzt, wird der Wert 1 im Flipflop gespeichert. Der Wert bleibt solange erhalten, bis der Rücksetzeingang r („reset“) auf 1 gesetzt wird, was zum Speichern einer 0 im Flipflop führt. Solange $r = s = 0$ ist, wird der gespeicherte Wert erhalten. Setz- und Rücksetzeingang dürfen beim RS-Flipflop nie gemeinsam auf 1 gesetzt werden. Ein **JK-Flipflop** hat zusätzlich zum Takt ebenfalls zwei Eingänge, die j und k genannt werden (die Namensgebung „JK“ ist nicht sicher überliefert; häufig wird es mit „jump“/„kill“ assoziiert). Es verhält sich wie ein RS-Flipflop für alle Eingangskombinationen, die bei diesem erlaubt sind, wobei j dem Eingang s entspricht und k dem Eingang r. Zusätzlich ist beim JK-Flipflop auch die Kombination $j = k = 1$ erlaubt, die dazu führt, dass der gespeicherte Wert negiert wird („toggle“). Ein **MS-Flipflop** besteht aus zwei zusammengeschalteten RS-Flipflops mit demselben Takt, die **Vorspeicherflipflop** (**VF**; „master“) und **Hauptspeicherflipflop** (**HF**; „slave“) genannt werden. Das MS-Flipflop hat neben dem Takteingang nur einen weiteren Eingang d. Ist $t = 1$, sind die Eingänge des HF gesperrt und d wird in das VF übernommen. Ist $t = 0$, sind die Eingänge des VF gesperrt und das HF übernimmt den Schaltzustand des VFs. Ein Einschreiben in den Speicher ist also nur möglich, wenn $t = 1$ gilt, ein Auslesen nur, wenn anschließend $t = 0$ gilt. Der Speicherinhalt kann bei allen beschriebenen Flipflop-Typen über den Ausgang q ausgelesen werden. Bei RS- und JK-Flipflops gibt es zusätzlich den Ausgang q' (auch $\overline{q}$), der den negierten Speicherinhalt liefert.

Weitere Informationen zu Schaltnetzen und Schaltwerken sowie speziell zu Flipflops können in der Vorlesungsaufzeichnung oder beispielsweise im Buch von Bernhard Eschermann gefunden werden [Esc93].

Themen dieses Kapitels

Die sieben Aufgaben dieses Kapitels beschäftigen sich mit dem theoretischen Aufbau von Schaltnetzen und Schaltwerken, der Erstellung von Schaltnetzen für bestimmte Funktionalitäten sowie deren Darstellung durch Wertetabellen und Boolesche Ausdrücke. Darüber hinaus wird die Erstellung von Schaltwerken über eine Definition der gewünschten Funktionalität durch endliche Automaten behandelt.

Dieses Thema in der Vorlesungsaufzeichnung zur Vorlesung „Grundlagen der Informatik II“:

`http://www.dasinfobuch.de/links/Schaltnetze-Schaltwerke.html`

Aufgaben

Aufgabe 1 CMO-AG

Schaltnetze ★⋆

a) Welche Aufgaben haben folgende zwei Konstrukte?

   1) Flipflop

   2) Register

b) Geben Sie an, wie man durch Verwendung von ausschließlich NOR-Gattern die folgenden Gattertypen darstellen kann.

   1) AND-Gatter

   2) OR-Gatter

c) Geben Sie an, wie man das XOR-Gatter ausschließlich durch Nutzung der folgenden Gattertypen darstellen kann.

   1) OR-, AND- und NOR-Gatter

   2) OR-, AND- und NOT-Gatter

→ *Lösung: S. 115*

Aufgabe 2 SCH-AA

Schaltnetze ★⋆

Ein Volladdierer realisiert auf Schaltkreisebene die Addition von zwei Eingabebits a, b und einem eventuell in früheren Additionen aufgetretenen Übertrag c. Als Ausgabe liefert er ein Summenbit S und ein Übertragsbit C für eventuelle künftige Additionen.

a) Geben Sie eine Wertetabelle für einen Volladdierer an.

b) Ermitteln Sie jeweils eine konjunktive Normalform (KNF) für die beiden Ausgänge. Wie weit kann man diese vereinfachen?

c) Geben Sie eine Realisierung des vereinfachten Schaltnetzes an, indem Sie ausschließlich 2-stellige AND-, OR- und XOR-Gatter verwenden. Legen Sie dabei für C die disjunktive Normalform (DNF) zugrunde und vereinfachen Sie diese.

→ *Lösung: S. 116*

Aufgabe 3 **SCH-AC**

Schaltnetze ★

Entwerfen Sie ein Schaltnetz, welches zwei vierstellige Dualzahlen

$$x = (d, c, b, a), x' = (d', c', b', a') \in \mathbb{B}^4$$

addiert und sich in Abhängigkeit der Einsen im Ergebnis folgendermaßen verhält: bei gerader Anzahl soll am Ausgang A eine 0 anliegen, bei ungerader Anzahl eine 1.

Hinweise:

- Für den Zahlenwert der Bitstrings gilt:

$$\text{wert}(x) = d \cdot 2^3 + c \cdot 2^2 + b \cdot 2^1 + a \cdot 2^0,$$
$$\text{wert}(x') = d' \cdot 2^3 + c' \cdot 2^2 + b' \cdot 2^1 + a' \cdot 2^0$$

- Benutzen Sie als Bausteine die üblichen Gatter AND, OR, NOT, XOR, ... sowie den Halbaddierer HA und den Volladdierer VA.
- Beachten Sie, dass auch der letzte Übertrag zum Ergebnis gehört.

→ *Lösung: S. 117*

Aufgabe 4 **SCH-AB**

Schaltnetze ★★

Eine Lampe wird von drei Schaltern s_1, s_2, und s_3 an- oder ausgeschaltet. Die Lampe soll leuchten, wenn

- s_1 aus, s_2 an, s_3 an, oder
- s_1 an, s_2 aus, s_3 an, oder
- s_1 an, s_2 an, s_3 an ist.

In allen anderen Fällen soll die Lampe nicht leuchten.

a) Stellen Sie eine Wahrheitstabelle auf, die beschreibt, wie sich der Zustand der Lampe L als Funktion $\mathbb{B}^3 \to \mathbb{B}$ von s_1, s_2, s_3 verhält. Dabei soll der Zustand „aus“ mit 0 und „an“ mit 1 kodiert werden.

b) Geben Sie die Schaltfunktion von L in disjunktiver Normalform (DNF) an.

c) Geben Sie die Schaltfunktion von L in konjunktiver Normalform (KNF) an.

→ *Lösung: S. 118*

Aufgabe 5 **SCH-AE**

Schaltwerke ★

a) Skizzieren Sie den prinzipiellen Aufbau eines getakteten Schaltwerkes.

b) Skizzieren Sie ein Schaltwerk, das genau dann eine 1 am Ausgang anliegen hat, wenn die bisher erfolgte Eingabe zwei aufeinander folgende Einsen enthält, also für Wörter der Sprache $L = \{\{0,1\}^m 11 \{0,1\}^n \mid m, n \in \mathbb{N}_0\}$. Benutzen Sie dafür nur AND-, OR- und NOT-Gatter sowie getaktete RS-Flipflops.

→ *Lösung: S. 118*

Aufgabe 6 **SCH-AF**

Schaltwerke ★

Skizzieren Sie ein Schaltwerk bei dem genau dann eine 1 am Ausgang a anliegt, wenn die am Eingang E bisher erfolgte Eingabefolge Element folgender Sprache ist:

$$L = \{w \in \{0,1\}^\star \mid |w|_1 \bmod 2 = 1\}$$

Hinweis: Das Schaltwerk erhält pro Takt t ein Signal 0 oder 1, die Länge dieser Eingabe ist unbegrenzt.

→ *Lösung: S. 120*

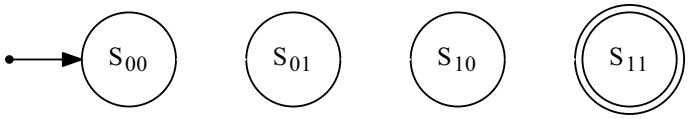
Aufgabe 7 **SCH-AD**

Schaltwerke ★★

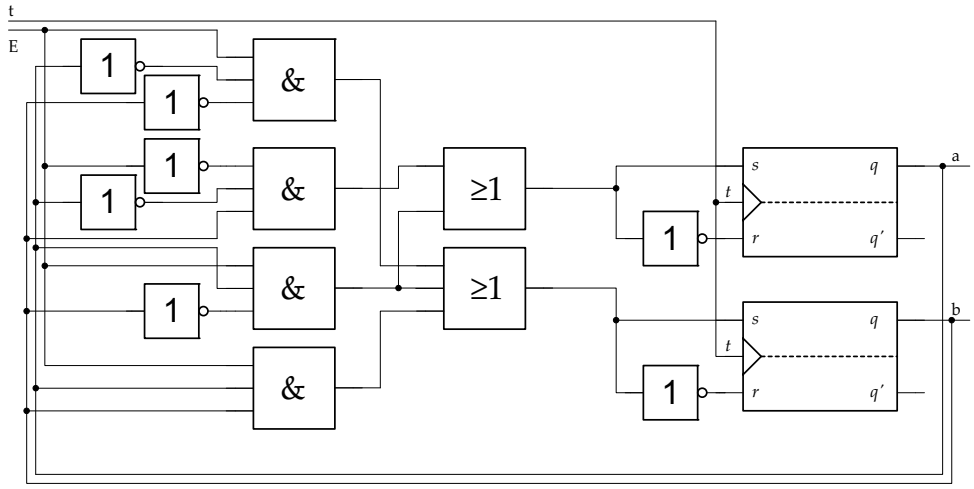
Gegeben seien der unvollständig definierte endliche Automat

$$A = (\{0,1\}, \{s_{00}, s_{01}, s_{10}, s_{11}\}, \delta, s_{00}, \{s_{11}\})$$

δ:



und das folgende Schaltwerk:



a) Vervollständigen Sie das Zustandsüberführungsdiagramm von A mithilfe des Schaltwerkes, sodass sich bei sukzessiver Abarbeitung eines Wortes $w \in \{0,1\}^\star$ (das über E in das durch t getaktete Schaltwerk eingegeben wird) der Automat genau dann in Zustand s_{ab} befindet, wenn die aktuelle Ausgabe des Schaltwerkes (a, b) ist, wobei $a, b \in \{0,1\}$.

b) Geben Sie einen regulären Ausdruck α an, sodass $L(\alpha) = L(A)$.

→ *Lösung: S. 120*

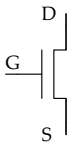
2 Complementary Metal Oxide Semiconductor (CMOS)

Einführung

Die CMOS-Technologie ist heute eine wichtige Grundlage für die Herstellung integrierter Schaltungen. Insbesondere Speicherelemente, Mikroprozessoren und Sensoren werden oft in CMOS-Technologie gefertigt. Dabei wird ein sogenannter **Halbleiter** (meist Silizium), also ein Stoff, dessen elektrische Leitfähigkeit zwischen der von Metallen und der von Isolatoren liegt, genutzt, um elektrische Schalter zu generieren, die über ein Eingangssignal in die beiden Zustände **leitend** („geschlossen“) und **nichtleitend** („offen“) geschaltet werden können. Solche Schalter bilden die physikalische Grundlage, um Gatter und Schaltnetze/Schaltwerke zu realisieren (vgl. Kapitel 1). Dabei werden Schalter unterschieden, die beim Anliegen einer „1“ (also einer positiven Spannung, die eine bestimmte Grenzspannung U_t überschreitet) am Schaltungseingang (Gate) leiten und sonst nicht leiten, und der dazu komplementären Variante, bei der beim Anliegen einer „0“ (also bei Unterschreitung der Grenzspannung) geleitet wird und sonst nicht. Das wesentliche Bauelement hochintegrierter Schaltungen ist dabei der Metal-Oxide-Semiconductor-Field-Effect-Transistor (MOSFET), auf dem der NMOS- und der PMOS-Transistor beruhen.

NMOS-Transistor

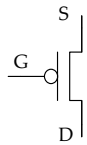
Der NMOS-Transistor repräsentiert die bei 1 leitende Variante der elektrisch steuerbaren Schalter in der CMOS-Technologie. Wir wollen hier nicht weiter auf die physikalischen Grundlagen und Eigenschaften von MOSFET-Transistoren eingehen (diese können in der Vorlesungsaufzeichnung studiert werden), sondern beschränken uns auf deren Funktionsweise auf der abstrakteren Ebene der Schaltungslogik. Der NMOS-Transistor verfügt über die drei Anschlüsse **Gate** G, **Source** S und **Drain** D, die im Schaltzeichen des NMOS-Transistors folgendermaßen angeordnet sind:



Dabei ist G der Schaltungseingang, dessen Spannung bestimmt, ob der Weg von D nach S durchlässig ist. Wenn bei G eine 1 anliegt, ist der Schalter geschlossen, also besteht eine elektrische Leitfähigkeit des Transistors zwischen D und S. Liegt bei G eine 0 an, ist der Schalter offen und der Transistor ist nicht durchlässig, also nichtleitend.

PMOS-Transistor

Der PMOS-Transistor verhält sich auf Ebene der Schaltungslogik exakt komplementär zum NMOS-Transistor. Wieder ist G der Schaltungseingang, dessen Spannung bestimmt, ob diesmal der Weg von S nach D durchlässig ist. Hier bedeutet jedoch eine 0 an G, dass sich der Schalter in einem durchlässigen oder leitenden Zustand befindet, also geschlossen ist, und eine 1 an G, dass er nichtdurchlässig, nichtleitend bzw. geöffnet ist. Das Schaltzeichen eines PMOS-Transistors sieht wie folgt aus:



CMOS als Kombination von NMOS- und PMOS-Transistoren

Während es grundsätzlich möglich ist, Schaltungen ausschließlich aus NMOS- oder ausschließlich aus PMOS-Elementen zu bauen, ist es vor allem in Bezug auf den Energieverbrauch vorteilhaft, beide Transistor-Typen gleichzeitig zu nutzen. Daher beruht die CMOS-Technologie auf einer Kombination von NMOS- und PMOS-Transistoren, die in zueinander **komplementärer** Weise mit der **positiven Spannung** V_{DD} (entspricht 1) und der **Masse** GND (entspricht 0) zu einer Schaltung verbunden werden. Dabei liegen alle PMOS-Elemente nahe an V_{DD} und alle NMOS-Elemente nahe an GND. Die Eingänge der Schaltung werden mit den Gate-Eingängen der PMOS- und NMOS-Elemente verbunden, sodass je nach Eingangsbelegung entweder von V_{DD} oder von GND aus eine leitende Verbindung mit dem Ausgang der Schaltung besteht. Der Ausgang wird dabei üblicherweise „in der Mitte“, also an den Drain-Verbindungen zwischen dem NMOS- und dem PMOS-Teil abgegriffen. Der Ausgang einer Schaltung kann wieder als Eingang einer neuen Schaltung oder eines Schaltungsteils dienen. Zu jedem NMOS-Transistor gehört also ein PMOS-Transistor, wobei sich die komplementären Transistoren gegenseitig insofern ergänzen müssen, als einerseits verhindert werden muss, dass in der Schaltung zu irgendeinem Zeitpunkt V_{DD} mit GND leitend verbunden wird (Kurzschluss), und andererseits gewährleistet sein muss, dass immer entweder von V_{DD} oder von GND aus eine leitende Verbindung zum Ausgang besteht (sonst unklares Ausgangssignal).

Die beiden Bereiche einer CMOS-Schaltung (PMOS/NMOS) können gemeinsam entworfen werden, da sich die Struktur der Verknüpfungen des einen Bereichs aus denen des anderen durch die komplementäre Bauweise ergeben. So kann man beispielsweise zunächst die PMOS-Schaltung entwerfen und erhält danach die zugehörige NMOS-Schaltung, indem für jede Parallelschaltung in PMOS eine Reihenschaltung in NMOS umgesetzt wird und umgekehrt. (Heute werden aus Effizienzgründen auch Schaltungen eingesetzt, die nicht in exakt dieser komplementären Weise aufgebaut sind; diese werden hier aber nicht betrachtet.)

Bemerkung: Studierende haben zu Beginn ihrer Beschäftigung mit der CMOS-Technologie häufig die etwas irreführende Vorstellung, dass, ähnlich wie bei der Darstellung von Schaltnetzen, Signale „von links nach rechts fließen“, also aus den Eingängen über die MOS-Elemente an die Ausgänge weitergeleitet werden. Auf einer abstrakten logischen Ebene ist das zwar richtig, hilfreicher ist es in diesem Zusammenhang allerdings, sich vorzustellen, dass die Belegung