

Digitaltechnik

H. Altmann, H. Blessing, H. Burkhardt und D. Pérard

Kurzbeschreibung

Digitale Geräte erscheinen auf den ersten Blick relativ kompliziert. Ihr Aufbau beruht jedoch auf dem einfachen Konzept der wiederholten Anwendung weniger Grundschaltungen. Die Verknüpfung dieser Grundschaltungen in einem Schaltnetz erhält man aus der Problemstellung durch Anwendung rein formaler Methoden. Die Hilfsmittel dazu liefert die Boolesche Algebra, die im speziellen Fall der Anwendung auf die Digitaltechnik als Schaltalgebra bezeichnet wird. Die Problemstellung ist meist in Form einer Funktionstabelle bzw. Wahrheitstafel gegeben. Die Aufgabe besteht zunächst darin, eine logische Funktion zu finden, die diese Funktionstabelle erfüllt. Im nächsten Schritt wird diese Funktion mit Hilfe eines Karnaugh-Veitch-Diagramms auf die einfachste Form gebracht. Anschließend kann man sie durch entsprechende Kombination der logischen Grundschaltungen realisieren.

Unter einem Schaltwerk versteht man eine Anordnung zur Durchführung logischer Verknüpfungen mit der zusätzlichen Fähigkeit, einzelne Variablenzustände zu speichern. Die Ausgangsvariablen hängen im Unterschied zum Schaltnetz nicht nur von den Eingangsvariablen, sondern zusätzlich von der Vorgeschichte der Ausgangs- bzw. Zwischengrößen ab. Die einfachsten Schaltwerke sind Kippstufen (Flip-Flops), deren Verknüpfung zu Speicherelementen, Zählern oder Schieberegistern führt.

Im ersten Abschnitt werden zunächst die Grundlagen der Schaltalgebra zusammengestellt und die Wirkungsweise von Kippstufen behandelt. Im zweiten Abschnitt werden konkrete Anwendungsbeispiele (eine Zählkette, die Steuerung einer Ampelanlage und ein Schieberegister) vorgestellt, die während des Praktikumsversuchs auf einem Steckbrett realisiert werden. Einen tieferen Einblick in diesen Themenbereich geben [Föl67, Hei71, Wey72, Beu92, Tie93].

Lernziele

1. Schaltalgebra: Grundverknüpfungen, Schaltnetze
2. Optimierungsverfahren mit Hilfe von KV-Diagrammen

Inhaltsverzeichnis

1 Grundlagen der Schaltalgebra	3
1.1 Einleitung	3
1.2 Definitionen	3
1.3 Die elementaren logischen Verknüpfungen	4
1.4 Weitere wichtige logische Verknüpfungen	4
1.5 Sätze und Rechenregeln für die logischen Verknüpfungen	6
1.6 Erstellen von Logikplänen aus logischen Gleichungen und umgekehrt . .	6
1.7 Normalformen der logischen Gleichungen	7
1.8 Optimierungsverfahren	9
1.9 Speicherelemente, Zählketten und Schieberegister	11
2 Versuchsaufbau und Aufgabenstellung	15
2.1 Zählerstopp	16
2.2 Steuerung einer Ampelanlage	17
2.3 Aufbau eines Schieberegisters	18
Literatur	19

1 Grundlagen der Schaltalgebra

1.1 Einleitung

In der Technik hat man es häufig mit Größen zu tun, zu deren eindeutiger Beschreibung die Angabe von zwei Zuständen genügt, z. B. „Schalter ein“ und „Schalter aus“ oder „Strom“ und „Kein Strom“ usw. Sind nun diese binären Zustandsgrößen in eindeutiger Weise durch einen logischen Zusammenhang miteinander verknüpft, so verwendet man zur mathematischen Beschreibung dieser funktionalen Abhängigkeit die Verknüpfungsregeln der Schaltalgebra. Die Schaltalgebra ist ein Begriff für die logische oder Boolesche Algebra, der bereits deren Anwendungszweck – die Berechnung technischer Schaltungen – zum Ausdruck bringt. Die Zustandsvariablen können hier nur binäre Werte annehmen. Die Boolesche Algebra selbst entstand aus der symbolischen Logik und wurde vom englischen Mathematiker G. Boole (1815-1846) entwickelt.

Die logischen Verknüpfungsglieder, welche die abhängige binäre Variable T (Ausgangsgröße) mit den unabhängigen Variablen A, B, C, \dots (Eingangsgrößen) verbinden, werden in der Technik mit zweiwertigen Schaltern realisiert. In der Anfangszeit verwendete man elektromagnetische Schalter (Relais), welche jedoch bald durch die wesentlich schnelleren und zuverlässigeren elektronischen Halbleiterschalter ersetzt wurden.

Ein typisches Anwendungsbeispiel der Schaltalgebra ist die automatische Steuerung eines Aufzugs. Hier wird die binäre Ausgangsvariable T mit zwei möglichen Zuständen „Aufzugsmotor ein, Bremse auf“ und „Aufzugsmotor aus, Bremse angezogen“ durch einen sinnvollen logischen Zusammenhang von mehreren zweiwertigen Eingangsvariablen wie z. B. „Tür auf“ und „Tür zu“ oder „Knopf gedrückt“ und „Knopf nicht gedrückt“ gesteuert.

Im vorliegenden Versuch sollen grundlegende Kenntnisse der Schaltalgebra vermittelt und anhand verschiedener praktischer Versuche vertieft werden.

1.2 Definitionen

Alle Variablen sollen binären Charakter haben. Der „ein“- oder „ja“- Zustand sei mit „1“, der andere mit „0“ bezeichnet, also z. B.

$$A = \begin{cases} 1 & \text{Signal vorhanden} \\ 0 & \text{Signal nicht vorhanden} \end{cases} \quad (1)$$

Der Übersichtlichkeit halber seien die System-Ausgangsvariable mit T oder Q und die Eingangsvariablen mit A, B, C, \dots bezeichnet. In einer zur Algebra analogen Weise läßt sich der Zusammenhang der Variablen A, B, C, \dots mit T durch eine Funktion mit der allgemeinen Form

$$T = F(A, B, C, \dots) \quad (2)$$

beschreiben. F bezeichnet man als Boolesche- oder auch als Schaltfunktion, die Variablen A, B, C, \dots, T, \dots als Schaltvariable oder Boolesche Variable.

1.3 Die elementaren logischen Verknüpfungen

Mit n binären Variablen A, B, C, \dots lassen sich 2^n voneinander verschiedene Kombinationen zu je n Elementen bilden (vergleiche mit n -stelligem Dualzahlensystem). Eine Schaltfunktion F kann in der Weise definiert werden, daß man zu allen möglichen Kombinationen der unabhängigen Variablen A, B, C, \dots den dazugehörigen Wert der abhängigen Variablen T in Form einer sogenannten Wahrheitstafel oder Wertetabelle angibt. Bild 1 zeigt die Wertetabelle für eine beliebige Schaltfunktion F_1 mit 2 Eingangsvariablen A, B . Die verschiedenen Kombinationen der beiden Variablen werden aus Gründen der Übersicht von oben nach unten in der Dualzahlenfolge aufgeschrieben. Der laufende Index i in der 1. Spalte bezeichnet die jeweilige Dualzahl in Dezimalschreibweise. Eine algebraische Darstellungsform der logischen Zusammenhänge

i	A	B	T
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

Abbildung 1: Wertetabelle.

läßt sich durch die Einführung der 3 logischen Grundverknüpfungen „UND“, „ODER“ und „NICHT“ erreichen, welche in der Booleschen Algebra durch folgende Kurzzeichen dargestellt werden: \wedge (UND), \vee (ODER) und $\overline{\dots}$ (NICHT). Diese logischen Operatoren werden durch die dazugehörigen Wertetabellen definiert (siehe Tabelle 1). Mit Hilfe dieser Symbole lassen sich die logischen Variablen ähnlich wie in der gewöhnlichen Algebra zu komplizierten logischen Gleichungen verknüpfen. Es läßt sich zeigen, daß jede beliebige Schaltfunktion mit Hilfe dieser 3 Grundverknüpfungen, ja sogar mit nur zwei dieser Operatoren (z. B. \wedge und $\overline{\dots}$) dargestellt werden kann.

Eine dritte Darstellungsmöglichkeit für einen logischen Zusammenhang bietet der Logikplan, bei dem eine logische Gleichung durch Verknüpfungen von Schaltsymbolen dargestellt wird. In Tabelle 1 sind die früher verwendeten runden Symbole und darunter die nach DIN 40700 genormten eckigen Symbole aufgeführt; letztere sind allerdings wenig überzeugend, und DIN 40700 wurde kürzlich zurückgezogen.

Die Tabelle 1 zeigt zudem eine Realisierung durch elementare Schaltglieder in einer Relaisschaltung für A und B im Zustand „0“.

1.4 Weitere wichtige logische Verknüpfungen

Weitere logische Verknüpfungen haben in der Praxis eigenständige Bedeutung erlangt, diese können aber stets aus den Grundverknüpfungen aufgebaut werden. Eine Zusammenstellung dieser Verknüpfungen zeigt Tabelle 2. Auch hier sind die Relaisschaltungen für den Zustand „0“ für A und B dargestellt.

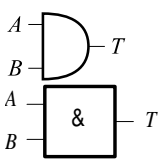
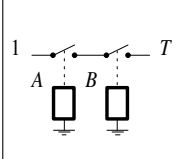
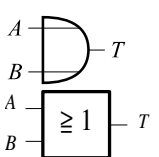
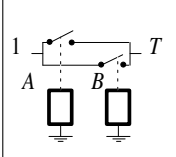
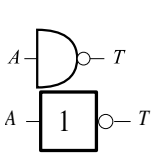
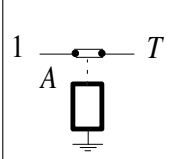
Bezeichnung	verbale Darstellung	Wertetab.	Gl.	Schaltsymb.	Relaisschalt.
- UND (AND) - logisches Produkt - Konjunktion	Die Ausgangsgröße entsteht genau dann, wenn <u>alle</u> Eingangsgrößen vorhanden sind.	$\begin{array}{ccc c} A & B & T \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{array}$	$\begin{array}{l} A \wedge B = T \\ 0 \wedge 0 = 0 \\ 0 \wedge 1 = 0 \\ 1 \wedge 0 = 0 \\ 1 \wedge 1 = 1 \end{array}$		
- ODER (OR) - logische Summe - Disjunktion	Die Ausgangsgröße entsteht genau dann, wenn <u>wenigstens eine</u> Eingangsgröße vorhanden ist.	$\begin{array}{ccc c} A & B & T \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{array}$	$\begin{array}{l} A \vee B = T \\ 0 \vee 0 = 0 \\ 0 \vee 1 = 1 \\ 1 \vee 0 = 1 \\ 1 \vee 1 = 1 \end{array}$		
- NICHT (NOT) - Komplementierung - Negation	Die Ausgangsgröße entsteht genau dann, wenn die Eingangsgröße <u>nicht</u> vorhanden ist.	$\begin{array}{cc c} A & T \\ \hline 0 & 1 \\ 1 & 0 \end{array}$	$\begin{array}{l} \bar{A} = T \\ \bar{0} = 1 \\ \bar{1} = 0 \end{array}$		

Tabelle 1: Die elementaren logischen Grundverknüpfungen.

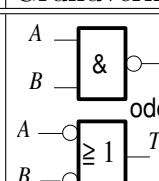
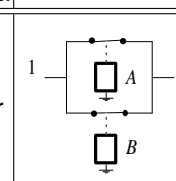
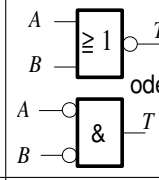
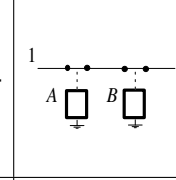
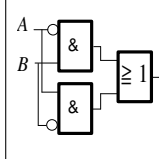
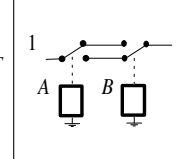
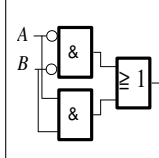
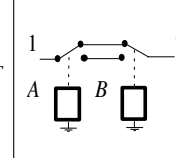
Bezeichnung	verbale Darstellung	Wertetabelle	Gl.	Aufbau aus Grundverkn.	Relaisschalt.
- NAND	Die Ausgangsgröße entsteht genau dann, wenn <u>mindestens eine</u> Eingangsgröße <u>nicht</u> vorhanden ist.	$\begin{array}{ccc c} A & B & T \\ \hline 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}$	$\begin{array}{l} T = \overline{A \wedge B} \\ T = \overline{A} \vee \overline{B} \end{array}$		
- NOR - WEDER-NOCH	Die Ausgangsgröße entsteht genau dann, wenn <u>alle</u> Eingangsgrößen <u>nicht</u> vorhanden ist.	$\begin{array}{ccc c} A & B & T \\ \hline 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{array}$	$\begin{array}{l} T = \overline{A \vee B} \\ T = \overline{A} \wedge \overline{B} \end{array}$		
- EXCLUSIVE OR - ENTWEDER-ODER - ANTIVALENZ	Die Ausgangsgröße entsteht genau dann, wenn <u>nur eine</u> der Eingangsgrößen vorhanden ist.	$\begin{array}{ccc c} A & B & T \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}$	$\begin{array}{l} T = (\overline{A} \wedge B) \vee (A \wedge \overline{B}) \end{array}$		
- ÄQUIVALENZ	Die Ausgangsgröße entsteht genau dann, wenn <u>alle</u> Eingangsgrößen <u>gleich</u> sind.	$\begin{array}{ccc c} A & B & T \\ \hline 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{array}$	$\begin{array}{l} T = (\overline{A} \wedge \overline{B}) \vee (A \wedge B) \end{array}$		

Tabelle 2: Weitere wichtige logische Verknüpfungen.

Die Verknüpfungen NAND und NOR haben in bestimmten Technologien eine relativ große Bedeutung.

1.5 Sätze und Rechenregeln für die logischen Verknüpfungen

Die in Tabelle 3 aufgelisteten Rechenregeln lassen sich einfach durch das Aufstellen der Wahrheitstafel beweisen.

Bezeichnung	Disjunktion (ODER-Verknüpfung)	Konjunktion (UND-Verknüpfung)
	$A \vee 0 = A$ $A \vee \bar{A} = 1$ $A \vee 1 = 1$ $A \vee A = A$	$A \wedge 1 = A$ $A \wedge \bar{A} = 0$ $A \wedge 0 = 0$ $A \wedge A = A$
Assoziativgesetz	$(A \vee B) \vee C = A \vee (B \vee C)$ $= A \vee B \vee C$	$(A \wedge B) \wedge C = A \wedge (B \wedge C)$ $= A \wedge B \wedge C$
Kommutativgesetz	$A \vee B = B \vee A$	$A \wedge B = B \wedge A$
Distributivgesetz	$(A \vee B) \wedge C = (A \wedge C) \vee (B \wedge C)$	$(A \wedge B) \vee C = (A \vee C) \wedge (B \vee C)$
De Morgan-Regel	$\overline{A \vee B} = \bar{A} \wedge \bar{B}$	$\overline{A \wedge B} = \bar{A} \vee \bar{B}$
Verschmelzungsgesetze	$A \vee (A \wedge B) = A$ $\bar{A} \vee (A \wedge B) = \bar{A} \vee B$	$A \wedge (A \vee B) = A$ $\bar{A} \wedge (A \vee B) = \bar{A} \wedge B$
Komplement	$\overline{\bar{A}} = A$	

Tabelle 3: Sätze und Rechenregeln.

1.6 Erstellen von Logikplänen aus logischen Gleichungen und umgekehrt

Zur Verdeutlichung der Vorgehensweise bedienen wir uns eines Beispiels. Gegeben sei die logische Gleichung

$$T = (A \vee B) \wedge [(\bar{B} \wedge C) \vee (\bar{A} \wedge B \wedge \bar{C})] . \quad (3)$$

Setzt man

$$\begin{aligned} T_1 &= A \vee B , \quad T_{21} = \bar{B} \wedge C , \quad T_{22} = \bar{A} \wedge B \wedge \bar{C} , \quad T_2 = T_{21} \vee T_{22} , \\ T &= T_1 \wedge [T_{21} \vee T_{22}] = T_1 \wedge T_2 ; \end{aligned} \quad (4)$$

so wird demnach T_{21} mit einem UND-Element mit einem direkten und einem negierten Eingang und T_{22} mit einem UND-Element mit einem direkten und zwei negierten Eingängen erzeugt. T_{21} und T_{22} werden mit einem ODER-Element mit zwei Eingängen zu T_2 verknüpft. T_1 wird mit einem ODER-Element erzeugt, und zusammen mit T_2

entsteht daraus über eine UND-Verknüpfung die gesuchte Variable T . Man baut auf diese Weise den Logikplan von den Eingangsgrößen her auf (s. Bild 2).

Soll die logische Gleichung aus dem Logikplan ermittelt werden, beginnt man am besten bei der Ausgangsvariablen T und gliedert dann schrittweise immer feiner auf:

$$T = T_1 \wedge T_2 = T_1 \wedge [T_{21} \vee T_{22}] = [A \vee B] \wedge [(\bar{B} \wedge C) \vee (\bar{A} \wedge B \wedge \bar{C})] . \quad (5)$$

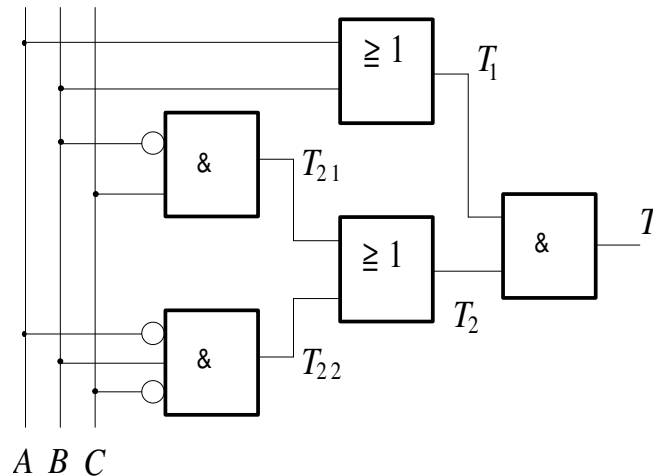


Abbildung 2: Logikplan zu Gl. (3).

1.7 Normalformen der logischen Gleichungen

Unter einem **Minterm** $m_i = F(A, B, C, \dots)$ von n Variablen versteht man eine Schaltfunktion von nur **konjunktiv** verknüpften Zustandsvariablen bzw. deren Negationen, mit der Eigenschaft:

$$m_i = \begin{cases} 1 & \text{genau für die } i\text{-te Dualbelegung der } n \text{ logischen Variablen} \\ 0 & \text{für alle anderen Belegungen .} \end{cases} \quad (6)$$

Diese Eigenschaft erhält man, indem man in der i -ten Dualbelegung die Konstanten „1“ durch die betreffende Variable und die Konstanten „0“ durch die jeweilige negierte Variable ersetzt und diese konjunktiv miteinander verknüpft. Da in einem Minterm alle vorhandenen Variablen miteinander verknüpft werden, wird er auch als **Vollkonjunktion** bezeichnet. An einem einfachen Beispiel mit 2 Variablen A, B wird dies für m_2 anhand der Wahrheitstafel in Bild 3 verdeutlicht. Jede beliebige Funktion F kann als **Disjunktion** von genau den **Mintermen** geschrieben werden, für die F in der entsprechenden Zeile den Wert „1“ annimmt. Diese spezielle Darstellung einer Schaltfunktion nennt man **disjunktive Normalform** oder auch **ODER-Normalform**. An dem folgenden Beispiel soll dies noch einmal gezeigt werden: In Bild 4 sei eine logische Funktion durch ihre Wahrheitstafel definiert. Die Ausgangsvariable nimmt den Wert

i	A	B	$F = m_2$
0	0	0	0
1	0	1	0
2	1	0	1
3	1	1	0

$$\Rightarrow m_2 = A \wedge \bar{B} \quad (7)$$

Abbildung 3: Wahrheitstafel und Minterm.

i	A	B	C	T
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

$$\begin{aligned} \Rightarrow T &= m_1 \vee m_2 \vee m_3 \\ &= (\bar{A} \wedge \bar{B} \wedge C) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge \bar{C}) \end{aligned} \quad (8)$$

Abbildung 4: Wertetabelle mit 3 Variablen.

„1“ genau für die 1., 2. und die 6. Dualbelegung an. Daraus erhält man die ODER-Normalform durch die disjunktive Verknüpfung der entsprechenden Minterme.

Unter einem **Maxterm** $M_i = F(A, B, C, \dots)$ versteht man die Negation des entsprechenden Minterms, d. h.:

$$M_i = \bar{m}_i = \begin{cases} 1 & \text{genau für die } i\text{-te Dualbelegung der logischen Variablen} \\ 0 & \text{für alle anderen Belegungen.} \end{cases} \quad (9)$$

Durch Negation des i -ten Minterms und Anwendung der De Morgan'schen Regeln erhält man somit das Bildungsgesetz für einen Maxterm. In der i -ten Dualbelegung wird die Konstante „0“ durch die dazugehörige Variable und die Konstante „1“ durch die Negation der betreffenden Variablen ersetzt und **disjunktiv** verknüpft. Bezüglich des Minterms m_2 aus Gl. (7) wird

$$M_2 = \bar{m}_2 = \overline{A \wedge \bar{B}} = \bar{A} \vee B. \quad (10)$$

Ähnlich der Bildung einer disjunktiven Normalform, läßt sich die **konjunktive Normalform** oder **UND-Normalform** erstellen. Hier werden alle Maxterme, für die die Funktion den Wert „0“ annimmt, konjunktiv miteinander verknüpft. Für das Beispiel aus Bild 4 erhält man somit eine äquivalente Darstellung der Ausgangsvariablen T in der konjunktiven Normalform durch:

$$\begin{aligned} T &= M_0 \wedge M_3 \wedge M_4 \wedge M_5 \wedge M_7 \\ &= (A \vee B \vee C) \wedge (A \vee \bar{B} \vee \bar{C}) \wedge (\bar{A} \vee B \vee C) \wedge (\bar{A} \vee B \vee \bar{C}) \wedge (\bar{A} \vee \bar{B} \vee \bar{C}). \end{aligned} \quad (11)$$

Zweckmäßigerweise wird man bei der Darstellung die Normalform wählen, welche die geringere Anzahl von Verknüpfungen ergibt. Das ist in dem Beispiel aus Bild 4 die disjunktive Normalform.

1.8 Optimierungsverfahren

Weder die konjunktive noch die disjunktive Normalform ist minimal in dem Sinne, daß sie in jedem Falle die geringste Zahl von Schaltelementen mit der geringsten Zahl von zu verknüpfenden Variablen ergibt. Zum Aufsuchen minimaler Formen gibt es Optimierungsverfahren. Die nächstliegende Methode zur Vereinfachung von Normalformen ist die Anwendung der in Abschnitt 1.5 aufgeführten Rechenregeln und Sätze. So wird aus der ODER-Normalform

$$T = (A \wedge B) \vee (A \wedge \bar{B}) \quad (12)$$

durch Anwendung des Distributivgesetzes

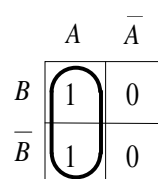
$$T = A \wedge (B \vee \bar{B}) . \quad (13)$$

Da $(B \vee \bar{B}) = 1$, wird $T = A$. Zur übersichtlichen Darstellung und systematischen Vereinfachung von ODER-Normalformen wurden von Karnaugh und Veitch die nach ihnen benannten KV-Diagramme entwickelt.

Das KV-Diagramm stellt für jeden Minterm aus den Variablen A, B, \dots einen Platz zur Verfügung, an den der Wert der Ausgangsvariablen T eingetragen wird. Benachbarte Plätze, welche die Eintragung „1“ besitzen, können hier zu Blöcken zusammengefaßt werden, welche lineare, rechteckige oder Quaderformen besitzen müssen. Ein Block wird durch die Konjunktion der Variablen bzw. deren Negation beschrieben, die für alle Elemente des Blockes gleich sind. Diese Blöcke werden dann disjunktiv verknüpft. Die einzelnen Blöcke dürfen sich überlappen und führen auf umso einfachere Ausdrücke, je größer sie sind. Bild 5(b) zeigt ein KV-Diagramm mit 2 Variablen A und B , welches sich aus der Wertetabelle aus Bild 5(a) erstellen läßt. Die ODER-Normalform, die sich

A	B	T
0	0	0
0	1	0
1	0	1
1	1	1

(a)



(b)

Abbildung 5: Wertetabelle (a) und KV-Diagramm (b).

aus der Tabelle in Bild 5(a) ergibt, lautet

$$T = (A \wedge B) \vee (A \wedge \bar{B}) . \quad (14)$$

Durch Zusammenfassen der beiden benachbarten Minterme im zugehörigen KV-Diagramm läßt sich auch schreiben:

$$T = A . \quad (15)$$

Bei nur zwei Variablen ist die Nachbarschaft von Plätzen sehr einfach zu erkennen (s. Bild 5(a)). Sind mehr als zwei Variablen vorhanden, so müssen die Nachbarschaftsbedingungen erweitert werden. Man kann sich das KV-Diagramm in allen Fällen als einen aufgeklappten Torus vorstellen. Damit sind alle gegenüberliegenden Plätze an den Rändern des Diagramms benachbart. Die Konstruktionsvorschrift für ein Diagramm mit n Variablen (n -ter Ordnung) kann wie folgt beschrieben werden:

Die Anzahl der Plätze im Diagramm wird pro hinzugekommener Ordnung verdoppelt. Dazu wird das Diagramm mit der Ordnung $n - 1$ an einer Seitenlinie gespiegelt. Dieses gespiegelte Diagramm wird zum ursprünglichen Diagramm an der Achse hinzugefügt. Die n -te Variable wird dem ursprünglichen, und deren Negation dem gespiegelten Diagramm zugeordnet. Die Spiegelachsen sind in Bild 6 und Bild 7 eingezeichnet.

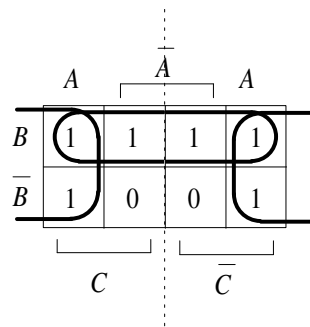


Abbildung 6: KV-Diagramm für 3 Variablen.

KV-Diagramm für 3 Variablen: Hier sind Blockgrößen von 2, 4 oder 8 Mintermen erlaubt. Plätze, die sich in der Längsrichtung gegenüberliegen, sind auch als benachbart zu betrachten. Die ODER-Normalform in nebenstehendem Beispiel besteht aus 6 Mintermen:

$$T = (A \wedge B \wedge C) \vee (\bar{A} \wedge B \wedge C) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge \bar{B} \wedge C) \vee (A \wedge \bar{B} \wedge \bar{C}) . \quad (16)$$

Durch Zusammenfassen der 6 mit 1 belegten Plätze zu 2 Vierer-Blöcken vereinfacht sich die Gleichung zu:

$$T = A \vee B . \quad (17)$$

KV-Diagramm für 4 Variablen : Hier können Blockgrößen von 2, 4, 8 oder 16 Mintermen gebildet werden. Sind, wie im Diagramm in Bild 7 eingezeichnet, alle 4 Eckplätze besetzt, so können auch diese zu einem Vierer-Block zusammengefaßt werden. Auch hier ergibt sich durch Bildung von 2 Vierer-Blöcken aus der ODER-Normalform mit 7 Mintermen ein einfacher Ausdruck:

$$T = (A \wedge B) \vee (B \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B} \wedge C \wedge \bar{D}) . \quad (18)$$

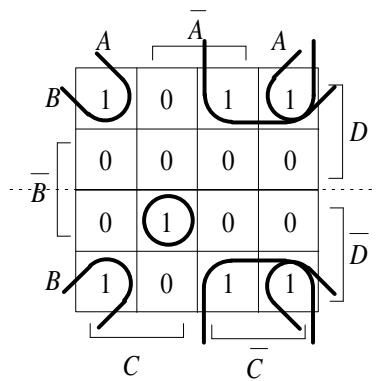


Abbildung 7: KV-Diagramm für 4 Variablen.

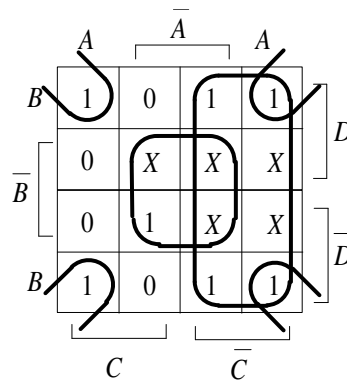


Abbildung 8: KV-Diagramm mit DON'T-CARES.

Bei manchen Anwendungen treten bestimmte Minterme nicht auf oder ihre Funktionswerte sind beliebig. Daher kann man ihnen im KV-Diagramm je nach Bedarf den Wert „1“ oder „0“ zuordnen und sie zur Blockbildung mitverwenden. Diese Plätze werden mit einem X gekennzeichnet und im Englischen als **DON'T-CARES** bezeichnet. Bild 8 zeigt ein KV-Diagramm, aus welchem sich die gleiche ODER-Normalform ergibt wie aus dem in Bild 7. Nur lassen sich die mit X gekennzeichneten Minterme zur Bildung eines Achter- und eines Vierer-Blockes mitverwenden. Damit vereinfacht sich Gl. (18) zu:

$$T = (A \wedge B) \vee (\bar{C}) \vee (\bar{A} \wedge \bar{B}) . \quad (19)$$

1.9 Speicherelemente, Zählketten und Schieberegister

Neben den logischen Grundverknüpfungen sind die bistabilen Kippstufen („Flip-Flops“) von großer Bedeutung. Im Gegensatz zu den bisher beschriebenen Verknüpfungsgliedern (welche auch Gatter genannt werden) besteht jetzt, wie wir sehen

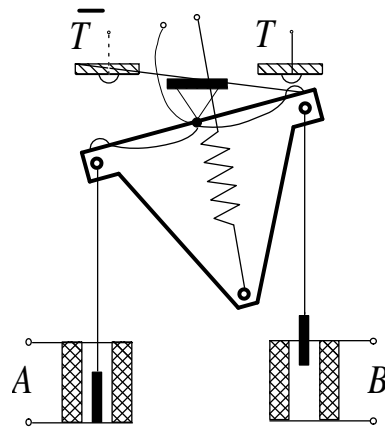


Abbildung 9: Mechanische Kippstufe.

werden, kein eindeutiger Zusammenhang mehr zwischen dem Zustand der Eingangsvariablen und der Ausgangsgröße T . Bild 9 zeigt eine mechanische Kippstufe. Neben T ist gestrichelt auch die komplementierte Ausgangsvariable \bar{T} dargestellt. In unserem Fall ist A die Stellgröße und B die Rückstellgröße. Die Arbeitsweise dieser Kippstufe läßt sich aus Bild 10 entnehmen, wobei dort willkürlich von dem Zustand $A = 0, B = 0$ und $T = 1$ zum Zeitpunkt t^0 ausgegangen wird. Der Zeitpunkt, unmittelbar nachdem eine der beiden Eingangsvariablen ihren Wert geändert hat, soll als neuer Beobachtungszeitpunkt der logischen Variablen gelten und in chronologischer Folge mit einem hochgestellten Index gekennzeichnet werden (A^n bedeutet die Variable A zum Zeitpunkt t^n). Der Buchstabe N in Zeile Nr. 7 in der Tabelle in Bild 10(a) bedeutet, daß diese Kombination der Zustände von A und B nicht auftreten darf, d. h. A und B der Bedingung $A \wedge B = 0$ gehorchen müssen, um ein richtiges Arbeiten des Speichergliedes zu gewährleisten. Bei Verknüpfungen von Gattern war die Ausgangsgröße eindeutig bestimmt durch die Eingangsgrößen zum gleichen Zeitpunkt, d. h.:

$$T^n = F(A^n, B^n, C^n, \dots) . \quad (20)$$

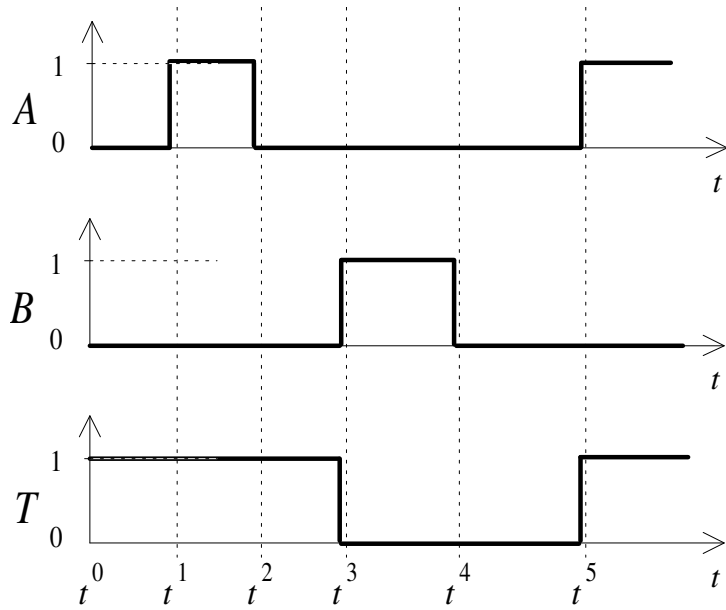
Eine logische Schaltung, welche nur Gatter und keine Rückführungen enthält, wird auch als **Schaltnetz** bezeichnet. Aus Bild 10(a) geht hervor, daß die Ausgangsgröße der Kippstufe nicht nur von den Eingangsgrößen zum gleichen Zeitpunkt, sondern auch noch zusätzlich von der Ausgangsgröße zum vorhergehenden Zeitpunkt abhängig ist. Die Kippstufe ist ein Schaltglied mit „Gedächtnis“ und kann prinzipiell mit einer Schaltfunktion der folgenden Form beschrieben werden:

$$T^n = F(A^n, B^n, T^{n-1}) . \quad (21)$$

Schaltungen, in denen neben Gattern auch Speicherelemente und/oder Rückführungen vorkommen, werden auch **Schaltwerke** genannt. Es ist leicht nachzuprüfen, daß die Kippstufe auch durch die Tabelle in Bild 11(a) beschrieben werden kann. Diese Kippstufe wird auch als **RS-Flip-Flop** (**R**eset/**S**et-Flip-Flop) bezeichnet. Das zugehörige

t	A	B	T
t^0	0	0	1
t^1	1	0	1
t^2	0	0	1
t^3	0	1	0
t^4	0	0	0
t^5	1	0	1
t^6	1	1	N
\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot

(a)

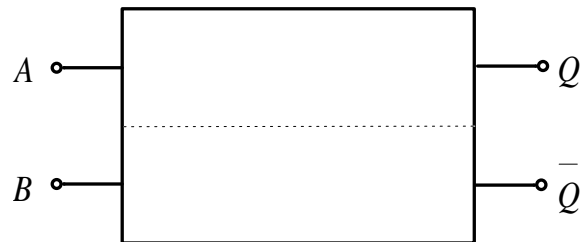


(b)

Abbildung 10: Wertetabelle (a) und Beobachtungszeitpunkte (b).

i	A^n	B^n	Q^{n-1}	Q^n
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	N
7	1	1	1	N

(a)



(b)

Abbildung 11: Wertetabelle (a) und RS-Flip-Flop (b).

Schaltsymbol zeigt Bild 11(b). A ist der **Set**- und B der **Reset**-Eingang. Sind die Eingänge A und B komplementär belegt, so übernimmt der Ausgang Q den Wert des Eingangs A .

Das hier beschriebene RS-Flip-Flop ist nicht getaktet. Das bedeutet, daß sich eine Änderung der Eingangssignale, bis auf eine technologieabhängige Reaktionszeit (Gat-

terlaufzeit), unmittelbar auf den Ausgangszustand auswirkt. Dies ist in vielen Fällen unerwünscht. Mit dem Ziel, alle Zustandsänderungen der Ausgänge synchron ablaufen zu lassen, werden taktgesteuerte Flip-Flops verwendet. Hier schaltet das Bauteil erst, wenn das Steuersignal (Takt) gegeben wird. Dies kann der Taktzustand (0,1) sein oder die Taktflanke, wenn der Pegel des Steuersignals von 0 auf 1 oder von 1 auf 0 umkippt.

Bei der Suche nach einem möglichst vielseitig einsetzbaren Flip-Flop ist man von einem taktflankengesteuerten RS-Flip-Flop ausgegangen. Daraus entstand das sogenannte JK-Flip-Flop. Bild 12(a) zeigt dessen Wahrheitstafel.

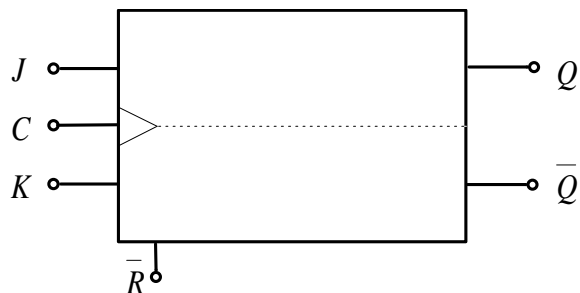
Das JK-Flip-Flop ist ähnlich aufgebaut wie das RS-Flip-Flop, nur besitzt es einen zusätzlichen Eingang C (Clock) für die Taktsteuerung und einen taktunabhängigen Reset-Eingang (\bar{R}). Außerdem hat die gleichzeitige Belegung der beiden Eingänge mit 1 einen wohldefinierten Sinn. Bild 12(b) zeigt das Symbol eines positiv flankengetriggerten JK-Flip-Flops. Der Triggermode (pos. Flanke) wird durch ein Symbol am Takteingang, hier ein Dreieck, gekennzeichnet. Durch unterschiedliche Belegung der Eingänge J und K lassen sich zwei wichtige Arten von Flip-Flops erzeugen: Das **T-**(Tast)-**Flip-Flop** und das **D-**(Delay)-**Flip-Flop**. Die Funktionsweisen der beiden Arten lassen sich wie folgt aus der Wertetabelle (Bild 12(a)) ablesen:

Im Fall 2 und 3 ist das D-Flip-Flop realisiert. Sind die Belegungen der Eingänge J und K komplementär, was sich sehr leicht mit einem Inverter realisieren läßt, so wird beim nächsten Takt der Wert des J-Einganges an den Ausgang Q weitergegeben.

Im Fall 4 liegt ein T-Flip-Flop vor. Sind beide Eingänge mit 1 belegt, ändert sich bei jedem Takt der Wert an den Ausgängen Q und \bar{Q} . Im Fall 1 bleiben die Werte der Ausgänge eingefroren. Mit T- und D-Flip-Flops lassen sich zwei elementare Bausteine der Digitaltechnik, nämlich Zählketten und Schieberegister, realisieren.

Fall	K^n	J^n	Q^{n+1}
1	0	0	Q^n
2	0	1	1
3	1	0	0
4	1	1	\bar{Q}^n

(a)



(b)

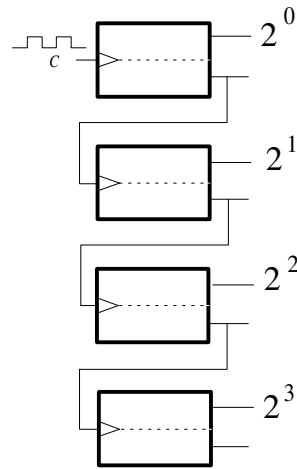
Abbildung 12: Wertetabelle (a) und taktflankengesteuertes (positive Flanke) JK-Flip-Flop (b).

Zählketten

Werden mehrere T-Flip-Flops wie in Bild 13(b) zu einer Kette verbunden, so ergibt sich ein dualcodierter Asynchrone Zähler. Gezählt wird die Anzahl der Pulse bzw. der Zustände 1 am Eingang, wobei das Ergebnis im Dualcode erscheint (s. Bild 13(a)). Jedes Speicherglied repräsentiert hier eine Dualstelle.

Zählerstand				
Dez.	Dualcode			
i	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
.

(a)



(b)

Abbildung 13: Zählerstandanzeige (dezimal und dual) (a) und Zähler aus T-Flip-Flops (b).

Schieberegister

Durch Verkettung von D-Flip-Flops können Schieberegister realisiert werden. Bild 14 zeigt den kompletten Aufbau eines 4-stufigen Schieberegisters, welches den Eingangszustand des 1. Flip-Flops (A) getaktet bis zum Ausgang des letzten Flip-Flops (Q_4) weiterleitet. Die Schiebegeschwindigkeit wird durch die Taktfrequenz von C bestimmt.

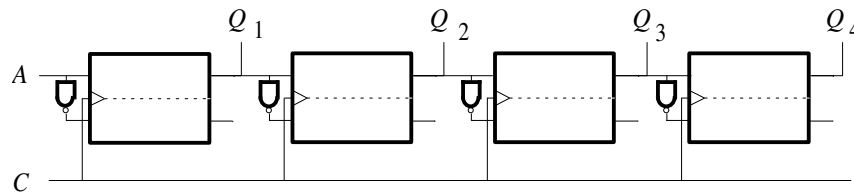


Abbildung 14: 4-stufiges Schieberegister aus D-Flip-Flops.

2 Versuchsaufbau und Aufgabenstellung

Zur Versuchsdurchführung wird ein Steckbrett bereitgestellt, auf welchem Gatter, Kippstufen und Taktgeneratoren integriert sind. Die Aufgaben sollen zunächst theoretisch mit Hilfe der beschriebenen Methoden bearbeitet werden. Die optimierten Lösungen sollen auf dem Steckbrett realisiert werden.

2.1 Zählerstopp

Es soll ein Zähler realisiert werden, welcher bei einem festlegbaren Zählerstand anhält und auf Befehl wieder von vorne startet.

Zähler: Der Zähler wird aus 4 T-Flip-Flops aufgebaut. Vorhanden sind nur JK-Flip-Flops welche als T-Flip-Flops arbeiten, solange die Eingänge J und K auf 1 liegen. Dies ist bei den verwendeten TTL-Bausteinen¹ auch dann gegeben, wenn die Eingänge offen sind. Sobald sie auf 0 gelegt werden, speichern die Flip-Flops ihren Zustand und der Zähler bleibt stehen.

Der Zählerstand wird auf der im Steckbrett integrierten 7-Segmentanzeige dezimal dargestellt.

Rückführung: Durch ein geeignetes Schaltnetz soll der vorgegebene Endstand des Zählers erkannt und ein Weiterzählen unterbunden werden. Bild 15 zeigt die Rückführung des Zählerstandes über ein Schaltnetz. Der Ausgang T der Schaltung belegt die J - und K -Eingänge der Zählerstufen mit 0 oder 1.

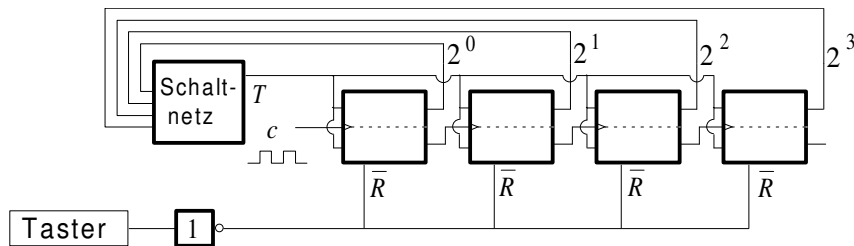


Abbildung 15: Zähler mit Rückführung über eine Stoplogik und Rücksetzmöglichkeit.

Das Schaltnetz muß mit Hilfe einer Wertetabelle und eines KV-Diagramms entworfen und optimiert werden. Für das folgende Beispiel soll der Zähler bei der Zahl 5 anhalten.

In der Wertetabelle in Bild 16 werden die Stellen des dualcodierten Zählerstandes als Eingangsvariablen eingetragen. Der Ausgangsvariablen T wird der Wert 1 für die Zahlen von 0 - 4 und der Wert 0 für die Zahl 5 zugewiesen. Alle anderen Zählerstände kommen in diesem Beispiel nicht vor und können daher als „DON'T CARES“ behandelt werden. Im KV-Diagramm in Bild 16 können die mit X gekennzeichneten Plätze zur Blockbildung mitverwendet werden. Es lassen sich zwei Achterblöcke bilden, wodurch sich T durch ein sehr einfaches Schaltnetz realisieren läßt:

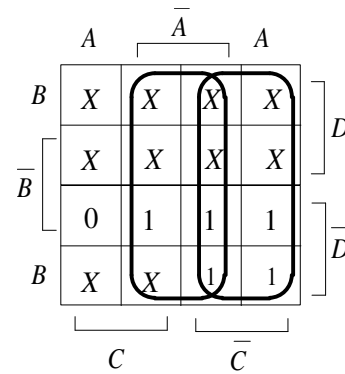
$$T^n = \bar{A} \vee \bar{C} . \quad (22)$$

Wird der Zählerstand über das so entwickelte Schaltnetz den J - und K -Eingängen zugeführt, stoppt der Zähler, sobald die Zahl 5 erreicht ist. Der Rücksetzvorgang wird manuell mit Hilfe eines Tasters realisiert, dessen Ausgangssignal über einen Inverter den globalen Rücksetzeingängen \bar{R} aller Zählerstufen gleichzeitig zugeführt wird (s. Bild 15).

¹TTL = Transistor-Transistor-Logik: Die Verknüpfungen werden hier ausschließlich durch bipolare Transistorsysteme realisiert [Beu92].

i	$2^3(D)$	$2^2(C)$	$2^1(B)$	$2^0(A)$	T
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	X
7	0	1	1	1	X

(a)



(b)

Abbildung 16: Wertetabelle (a) und KV-Diagramm (b) für einen Zählerstopp bei $i = 5$.

2.2 Steuerung einer Ampelanlage

Es soll eine Steuerung entworfen und auf dem Steckbrett realisiert werden, die eine Ampelanlage, bestehend aus zwei Ampeln, betreiben kann. Es wird zu Beginn der Versuchsdurchführung ein Zeitschema vorgegeben, nach dem die beiden Ampeln arbeiten sollen. Darin ist festgelegt, welchen Zustand die Ampeln in Abhängigkeit von einem Zählerstand annehmen sollen, siehe beispielsweise Bild 17.

Zunächst soll ein 4-Bit-Zähler wie in Versuch 2.1 aus 4 T-Flip-Flops aufgebaut und mit der 7-Segmentanzeige verbunden werden. Darüberhinaus soll beim Auftreten der Zahl 10 ein Schaltnetz die Flip-Flops automatisch über deren \bar{R} -Eingänge zurücksetzen. Dieses Schaltnetz soll mit Hilfe der hier behandelten Verfahren entworfen werden.

Der Entwurf der Steuerung wird für Ampel 1 anhand des Zeitschemas in Bild 17 und den KV-Diagrammen in Bild 18 exemplarisch vorgeführt. Aus den Diagrammen in Bild 18 lassen sich für die Ampelphasen Rot_1 und $Gelb_1$ die folgenden Gleichungen erarbeiten:

$$Rot_1 = (\bar{C} \wedge \bar{D}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{D}), \quad (23)$$

$$Gelb_1 = (A \wedge D) \vee (\bar{A} \wedge \bar{B} \wedge C). \quad (24)$$

Die Funktion für $Grün_1$ läßt sich hier sehr einfach durch

$$Grün_1 = \overline{Rot_1} \wedge \overline{Gelb_1}. \quad (25)$$

darstellen, da die Ampelphase Grün nur dann vorhanden sein darf, wenn Rot und Gelb nicht vorhanden sind (s. Bild 17).

Für Ampel 2 lassen sich auf gleichem Wege die folgenden Funktionen erstellen:

$$Rot_2 = D \vee (A \wedge C) \vee (B \wedge C), \quad (26)$$

$$Gelb_2 = (A \wedge D) \vee (\bar{A} \wedge \bar{B} \wedge C), \quad (27)$$

$$Grün_2 = \overline{Rot_2} \wedge \overline{Gelb_2}. \quad (28)$$

Zählerstand					Ampel 1			Ampel 2			\bar{R}
i	$2^3(D)$	$2^2(C)$	$2^1(B)$	$2^0(A)$	Rot_1	$Gelb_1$	$Grün_1$	Rot_2	$Gelb_2$	$Grün_2$	
0	0	0	0	0	1	0	0	0	0	1	1
1	0	0	0	1	1	0	0	0	0	1	1
2	0	0	1	0	1	0	0	0	0	1	1
3	0	0	1	1	1	0	0	0	0	1	1
4	0	1	0	0	1	1	0	0	1	0	1
5	0	1	0	1	0	0	1	1	0	0	1
6	0	1	1	0	0	0	1	1	0	0	1
7	0	1	1	1	0	0	1	1	0	0	1
8	1	0	0	0	0	0	1	1	0	0	1
9	1	0	0	1	0	1	0	1	1	0	1
10	1	0	1	0	X	X	X	X	X	X	0
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

Abbildung 17: Zeitschema für eine Ampelsteuerung.

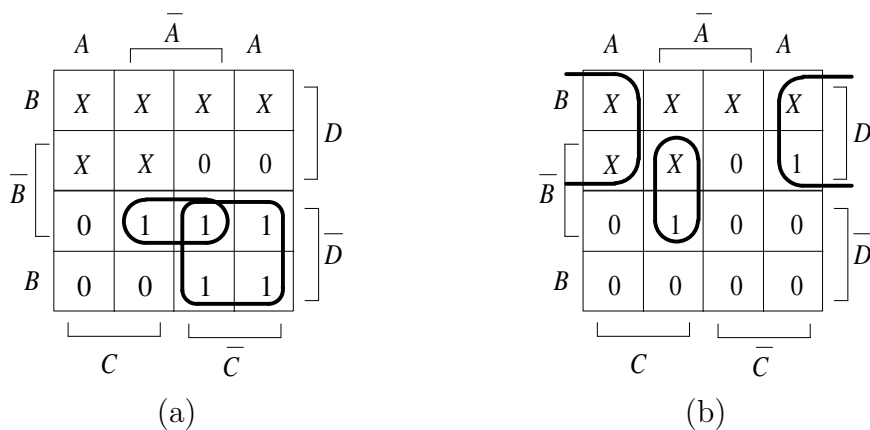


Abbildung 18: KV-Diagramme für Rot_1 (a) und $Gelb_1$ (b).

2.3 Aufbau eines Schieberegisters

Aus den auf dem Steckbrett vorhandenen JK-Flip-Flops soll nach Bild 14 ein 4-stufiges Schieberegister aufgebaut werden. Die Funktion soll mit Hilfe des Tasters überprüft werden, indem ein Signal auf den Eingang des ersten Flip-Flops aufgebracht wird. Das Schieberegister muß dieses Signal bis zum letzten Flip-Flop weitergeben.

Literatur

- [Föl67] O. Föllinger, W. Weber: *Methoden der Schaltalgebra*. R. Oldenburg Verlag, Berlin, 1967.
- [Hei71] K. Heim: *Schaltungsalgebra*. 3. Auflage, Verlag Siemens A.G., München, 1971.
- [Wey72] U. Weyh: *Elemente der Schaltungsalgebra*. 7. Auflage, R. Oldenburg Verlag, München, 1972.
- [Beu92] K. Beuth: *Elektronik 4 - Digitaltechnik*. 9. Auflage, Vogel-Buchverlag, Würzburg, 1992.
- [Tie93] U. Tietze, C. Schenk: *Halbleiter-Schaltungstechnik*. 10. Auflage, Springer Verlag, Berlin, 1993.