Digitale Systeme

vormals "Technische Grundlagen der Informatik 1"

Kapitel 2: Logik und Boolesche Algebra

basierend auf den Unterlagen der Vorlesung von
Prof. Dr.-Ing. Carsten Gremzow

Prof. Dr.-Ing. Sebastian Möller

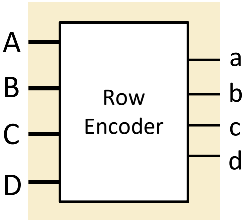
Technische Universität Berlin
Quality and Usability Lab
Institut für Theoretische Informatik und Softwaretechnik
Fakultät IV – Elektrotechnik und Informatik

Inhalt Digitale Systeme

1. Motivation und Einführung
2. Logik und Boolesche Algebra
   - Digital vs. Analog
   - Grundbegriffe der Logik
   - Logische Operationen
   - Boolesche Ausdrücke und Aussagen
   - Logische Funktionen, Bündel
3. Gatter und CMOS-Technik
4. Zahlendarstellung und Codes
5. Schaltnetze und Normalformen
6. Optimierung von Schaltnetzen
7. Standard-Schaltnetze
8. Speicherelemente und programmierbare Logik
9. Synchrone Schaltwerke
10. Register-Transfer-Entwurf und Mikroprogrammierung
11. Anwendungen Digitaler Systeme

Zentrale Fragen Kapitel 2

<u>Spezifikation</u> der Funktion von Hardware



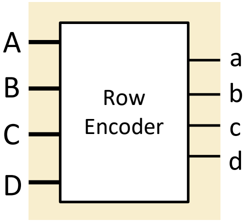
z.B. als logische Funktionen

$$a = D + \overline{A}\,\overline{C} + AB$$
$$b = \overline{C} + AB + \overline{A}\;\overline{B}$$
$$c = \;\ldots$$

… oder Wahrheitstabellen

Input				Output			
D	C	B	A	a	b	c	…
0	0	0	0	1	1	1	…
0	0	0	1	0	1	1	…
0	0	1	0	1	1	0	…
0	0	1	1	1	1	1	…

Zentrale Fragen Kapitel 2



- Was heißt „digital“?
- Was ist eine logische Operation?
- Wie rechnet man mit Boolescher Algebra?
- Gibt es mehrstellige logische Operationen?

z.B. als logische Funktionen

$a = D + \overline{A}\,\overline{C} + AB$

$b = \overline{C} + AB + \overline{A}\,\overline{B}$

$c = \ \ldots$

… oder Wahrheitstabellen

Input				Output			
D	C	B	A	a	b	c	…
0	0	0	0	1	1	1	…
0	0	0	1	0	1	1	…
0	0	1	0	1	1	0	…
0	0	1	1	1	1	1	…

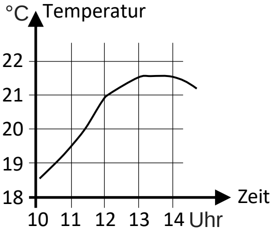
2.1 Digital vs. Analog

IBM Personal Computer Model 5150 - 1981

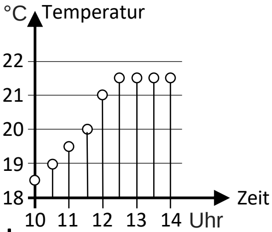
2.1 Digital vs. Analog

Diskretisierung der Welt

Physikalische Größen mit kontinuierlichem Wertebereich.
-> Unendlich viele Werte werden durch endlich viele ersetzt; dazu:
Rasterung, Diskretisierung -> zeitdiskret, wertdiskret.



a



b

Zeit	°C
10.00	18.5
10.30	19.0
11.00	19.5
11.30	20.0
12.00	21.0
12.30	21.5
13.00	21.5
13.30	21.5
14.00	21.5

c

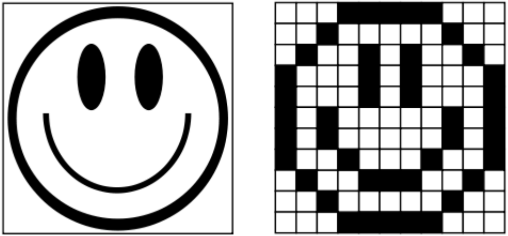
a analoge (zeit- und wertkontinuierlich)
b diskretisierte (zeit- und wertdiskret)
c digitalisierte Darstellung (codiert in Symbolfolgen)

Diskretisierung vs. Codierung

- Die **Digitalisierung** erfordert als Zwischenschritt also die **Diskretisierung**! Die diskrete Darstellung ist im Wesentlichen aber immer noch analog, d.h., der höheren Temperatur entspricht die längere Strecke!
  - **analog** = Abbildung der stetigen/kontinuierlichen Größen Zeit und Temperatur auf die ihnen analoge Größe “Länge einer Strecke”
  - **digital** = ziffernhaft, d.h. **Codierung** von endlich vielen Werten der diskretisierten Wertebereiche durch Symbolfolgen.
- Im Rechner sind alle Daten durch Binärcodierung digital dargestellt!

2.1 Digital vs. Analog

Beispiel



0	0	0	1	1	1	1	1	0	0	0
0	0	1	0	0	0	0	0	1	0	0
0	1	0	0	1	0	1	0	0	1	0
1	0	0	0	1	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0	1
1	0	1	0	0	0	0	0	1	0	1
1	0	1	0	0	0	0	0	1	0	1
1	0	0	1	0	0	0	1	0	0	1
0	1	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0
0	0	0	1	1	1	1	1	0	0	0

- Die Zeichnung ist nach der Rasterung (Diskretisierung) direkt mit den Symbolen 0 und 1 codierbar (Digitalisierung).
- Bei farbigen Zeichnungen lassen sich Farbe und Farbintensität durch Zahlen codieren.

2.1 Digital vs. Analog

Genauigkeit

Die digitale Darstellung kann man beliebig genau machen!
Maßgeblich ist die Anzahl an Bits.

Verlustlose Speicherbarkeit

Anders als bei kontinuierlich veränderlichen Werten gibt es beim Speichern und späterem Lesen von Binärzeichen keinen Genauigkeitsverlust.

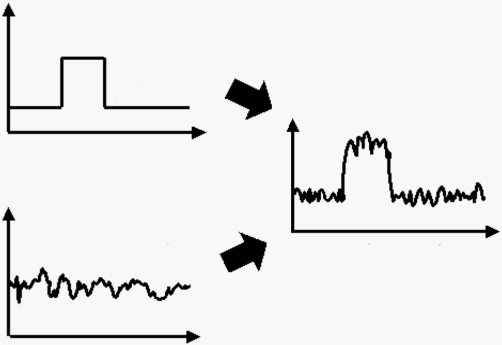
Geringe Störempfindlichkeit

Binäre Symbole können durch zweiwertige physikalische Größen dargestellt werden (Alles-oder-Nichts-Prinzip!)
Es reicht die Unterscheidbarkeit von z.B. Spannung und keine Spannung (auf den Spannungswert kommt es nicht an).

2.1 Digital vs. Analog

Beispiel

Geringe Störempfindlichkeit



2.1 Digital vs. Analog

Inhalt Digitale Systeme

Aussagen

- Jede wörtlich formulierte Behauptung ist eine Aussage (z.B. “3 + 5 = 8",“er läuft").
- Von solchen Sätzen lässt sich im Allgemeinen bestimmen, ob sie zutreffen oder nicht, ...
- ... d.h. ob sie **wahr** oder **falsch** sind.
- Es lässt sich somit der Wahrheitswert einer Aussage bestimmen.

Definition

Ein Satz, dem eindeutig ein Wahrheitswert zugeordnet werden kann, heißt Aussage. Ist A eine Aussage, dann ist W(A) der Wahrheitswert der Aussage mit: W(A) := wahr (falsch), falls Aussage A zutrifft (falls Aussage A nicht zutrifft).

2.2 Grundbegriffe der Logik

Verknüpfungen von Aussagen

- Mehrere Aussagen lassen sich zu einer neuen Aussage zusammenfassen und ...
- ... auch dafür lässt sich ein Wahrheitswert finden.
- Die Zusammenfassung von Einzelaussagen wird umgangs- sprachlich mit Hilfe von Bindewörtern gebildet.

Beispiel

5 ist eine Primzahl <u>und</u> 10 ist durch 5 teilbar.
Ist a durch 4 teilbar, <u>dann</u> ist a auch durch 2 teilbar.

Verknüpfungen von Aussagen

- Allgemein für A1 und A2 als zwei Aussagen gilt:

A1 und A2	Konjunktion
A1 oder A2	(einschließendes) Oder, Disjunktion
entweder A1 oder A2	ausschließendes Oder, Antivalenz
wenn A1 dann A2	Folgerung, Implikation
A1 genau dann, wenn A2	Äquivalenz

- Bindewörter verknüpfen Aussagen.
- Jeder dieser aus Aussagen zusammengesetzte Satz ist selbst wieder eine Aussage.
- Die Wahrheitswerte der Einzelaussagen stehen dem Wahrheitswert der zusammengesetzten Aussagen gegenüber.

Wahrheitswert der Verknüpfung

- Aussagenlogik betrachtet nur den Wahrheitswerte einer Aussage.
- Inhalt der Aussage wird vernachlässigt.
- Für Verknüpfungen zwischen den Wahrheitswerten gilt:

Verknüpfung der Wahrheitswerte der Einzelaussagen =
Wahrheitswert der zusammengesetzten Aussage

Beispiel

Ist z. B. "A1 und A2" eine zusammengesetzte Aussage, dann soll gelten:

W(A1) und W(A2) = W(A1 und A2)

Diese Gleichung muss für alle möglichen Kombinationen der Wahrheitswerte von A1 und A2 gelten.

Wertetabelle, Wahrheitstabelle

- “A1 und A2” ist nur dann eine wahre Aussage, wenn beide Einzelaussagen A1 und A2 zutreffen.
- Für die Verknüpfung lässt sich eine Wertetabelle bzw. Wahrheitstabelle aufstellen.

Beispiel

Wertetabelle der Konjunktion oder UND-Verknüpfung (Symbol: ∧):

W (A1)	W(A2)	W(A1) ∧ W(A2)
falsch	*falsch*	*falsch*
falsch	*wahr*	*falsch*
wahr	*falsch*	*falsch*
wahr	*wahr*	*wahr*

2.2 Grundbegriffe der Logik

Vereinfachung - Wertetabelle, Wahrheitstabelle

- Wahr und falsch wird durch w und f ...
- ... oder durch 1 und 0 dargestellt.
- w und f sind innerhalb der *Aussagenlogik* üblich ...
- ... 1 und 0 in der *formalen Logik*.

Beispiel

Wertetabelle der Konjunktion oder UND-Verknüpfung (Symbol: ∧):

W(A1)	W(A2)	W(A1) ∧ W(A2)
0	*0*	*0*
0	*1*	*0*
1	*0*	*0*
1	*1*	*1*

Inhalt Digitale Systeme

1. Motivation und Einführung
2. Logik und Boolesche Algebra
   - ✓ Digital vs. Analog
   - ✓ Grundbegriffe der Logik
   - ❑ Logische Operationen
   - ❑ Boolesche Ausdrücke und Aussagen
   - ❑ Logische Funktionen, Bündel
3. Gatter und CMOS-Technik
4. Zahlendarstellung und Codes
5. Schaltnetze und Normalformen
6. Optimierung von Schaltnetzen
7. Standard-Schaltnetze
8. Speicherelemente und programmierbare Logik
9. Synchrone Schaltwerke
10. Register-Transfer-Entwurf und Mikroprogrammierung
11. Anwendungen Digitaler Systeme

Logische Operationen

- Analog zur Wertetabelle für die Konjunktion, lassen sich Tabellen für alle anderen zusammengesetzten Aussagen aufstellen.
- Diese Verknüpfungen werden logische Operationen genannt.
- Anzahl der Wahrheitswerte oder Operanden bestimmt Operationsgrad
  - 1- oder 2-stellige Operationen
  - im Allgemeinen n-stelligen Operationen
  - 0-stellige Operationen?
- Die Konjunktion ist nur ein Beispiel für eine 2-stellige Operation
- **Quizfrage:** Wieviele 2-stellige, logische Operationen gibt es?

Logische Operationen

- In einer Wertetabelle für zwei Operanden kann jede beliebige Kombination von 0-1-Werten eingesetzt werden.
- Nicht alle Operationen haben einen Namen oder eine symbolische Bezeichnung.
- Es ergeben sich $2^{2^2} = 16$ Möglichkeiten.

a	b		$a \wedge b$		a		b	$a \not\equiv b$	$a \vee b$	$\overline{a \vee b}$	$a \equiv b$	$\bar{b}$		$\bar{a}$	$a \to b$	$\overline{a \wedge b}$	
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Elementare logische Operationen

Negation (NOT): $\neg a, \overline{a},$ "nicht a"

a	$\neg a$
0	1
1	0

Konjunktion (AND): $a \wedge b$, $a \cdot b$, "a und b"

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

Elementare logische Operationen

Disjunktion (OR): $a \vee b$, $a + b$, „a oder b“

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Äquivalenz (XNOR): $a \equiv b$, $\overline{a \oplus b}$, $a \leftrightarrow b$, „a äquivalent b“

a	b	$a \equiv b$
0	0	1
0	1	0
1	0	0
1	1	1

Elementare logische Operationen

Antivalenz (XOR): $a \not\equiv b$, $a \oplus b$, "a antivalent b"

a	b	$a \not\equiv b$
0	0	0
0	1	1
1	0	1
1	1	0

Implikation: $a \rightarrow b$, "a impliziert b"

a	b	$a \rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

Elementare logische Operationen

Sheffer-Funktion (NAND): $\overline{a \wedge b}$, $\overline{a \cdot b}$, $a|b$, „a und b nicht“

a	b	$\overline{a \wedge b}$
0	0	1
0	1	1
1	0	1
1	1	0

Peirce-Funktion (NOR): $\overline{a \vee b}$, $\overline{a + b}$, $a \downarrow b$, „weder a noch b“

a	b	$\overline{a \vee b}$
0	0	1
0	1	0
1	0	0
1	1	0

Elementare logische Operationen

Nochmal zur Implikation...

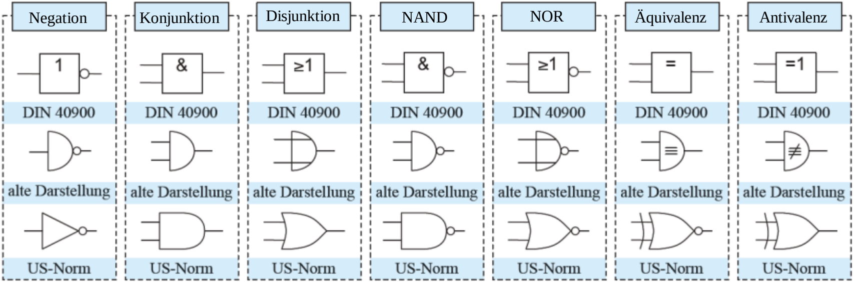
Beispiel

Die Implikation zu

- a *"Lenkung nicht kaputt"*
- b *"Fahrer betrunken"*
- $a \rightarrow b$ *"Auto fährt Zick-Zack"*
  - $a = 1, b = 1, 1 \rightarrow 1 = 1$
  - $a = 0, b = 1, 0 \rightarrow 1 = 1$
  - $a = 1, b = 0, 1 \rightarrow 0 = 0$
  - $a = 0, b = 0, 0 \rightarrow 0 = 1$

2.3 Logische Operationen

... als Gattersymbolik



Symolische Darstellung verschiedener Logikgatter. Aus D.W. Hoffmann, 2014, Abb. 5.16.

n-stellige Operationen

- Verknüpfungen von nur zwei Operanden sind im Allgemeinen nicht ausreichend.
- Bei mehrstelligen Operationen wächst die Zahl der möglichen Verknüpfungen rasant an.
    - Bei n-stelligen Operationen ergeben sich 2^n verschiedene Belegungen und ...
    - … es können wieder 2^{2^n} 0-1-Kombinationen als Ergebnisse zugeordnet werden.
- Alle mehrstelligen Operationen lassen sich durch 1- und 2-stellige Operationen ausdrücken.
- Die Grundverknüpfungen genügen, um sämtliche weiteren logischen Zusammenhänge zu beschreiben.
- Kombiniert man verschiedene Operationen erhält man **Ausdrücke**.

n-stellige Operationen

Definition

Eine Verknüpfung von endlich vielen Konstanten und logischen Variablen mittels Grundverknüpfungen heißt ein Ausdruck. Die Reihenfolge, in der Operationen anzuwenden sind, wird durch Klammern bestimmt. Ausdrücke ,die nur aus den Operationen $\overline{}$ *(Negation),* $\cdot$ *(Konjunktion) und* $+$ *(Disjunktion) zusammengesetzt sind, heißen boolesche Ausdrücke.*

n-stellige Operationen

Beispiel

Ausdrücke

$$\bar{a}$$

$$(a \cdot b) + 1$$

$$(a \rightarrow b) \cdot (\overline{c + a})$$

$$a \rightarrow b \oplus c$$

- *ist kein gültiger Ausdruck*
- *Reihenfolge der Operationen ist nicht eindeutig!*

2.3 Logische Operationen

Ermittlung des Wahrheitswerts eines Ausdrucks

- aktuellen Variablenwerte werden in den Ausdruck eingesetzt
- schrittweise Auswertung entsprechend der Klammerung
- jeweils die Grundverknüpfungen durch den Wahrheitswert ersetzen

Beispiel

Mit a =1, b =0, c =1 *gilt für den Wahrheitswert des Ausdrucks*

$(a \rightarrow b) \cdot (\overline{c + a})$:

$(a \rightarrow b) \cdot (\overline{c + a}) = (1 \rightarrow 0) \cdot (\overline{1 + 1}) = 0 \cdot 0 = 0$

Tipp: Soll der Ausdruck vollständig durch eine Wertetabelle beschrieben werden, ist es meist sinnvoll, auch Zwischenschritte mit in der Tabelle zu notieren!

Ermittlung des Wahrheitswerts eines Ausdrucks

Beispiel

Die Verknüpfung $(a \rightarrow b) \cdot (\overline{c + a})$ *soll durch eine Wertetabelle vollständig beschrieben werden:*

a	b	c	$(a \rightarrow b)$	$(\overline{c + a})$	$(a \rightarrow b) \cdot (\overline{c + a})$
0	0	0	1	1	1
0	0	1	1	0	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	0	0
1	1	1	1	0	0

Äquivalente Ausdrücke

Definition

Zwei Ausdrücke A1, A2 heißen äquivalent (A1 = A2) genau dann, wenn sie bei gleicher Belegung gemeinsamer Variablen stets gleiche Wahrheitswerte annehmen. A1 = A2 heißt dann eine logische Gleichung.

Dabei ist nicht gefordert, dass beide Ausdrücke genau dieselben Variablen enthalten.

Äquivalente Ausdrücke

Beispiel

Seien a, b, c *logische Variable.*
Behauptung: $a + \left(b \cdot \bar{b}\right) = a + (c \cdot \bar{c})$, *Beweis durch Wertetabelle:*

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

Beide Ausdrücke hängen weder von b noch von c ab.

Vorrangregeln

- Vermeidung zahlreicher Klammern durch Vorrangregeln
- analog zu „Punktrechnung vor Strichrechnung”

Definition

§1 *Geklammerte Verknüpfungen haben Vorrang vor Negation ($\overline{\ }$)*

§2 *Negation hat Vorrang vor Konjunktion ($\cdot$)*

§3 *Konjunktion hat Vorrang vor allen anderen Operationen*

§4 *Negationszeichen ($\overline{\ }$) über mehrere Variable oder Konstante, gelten diese als geklammert und die Negation gilt für den Klammerausdruck.*

§5 *mehreren aufeinanderfolgenden zweistelligen Grundverknüpfungen gleicher Vorrangstufe, werden von links nach rechts abgearbeitet*

§6 *Das Konjunktionszeichen kann weggelassen werden, wenn keine Verwechslungen mit anderen Variablen möglich sind.*

Vorrangregeln

Beispiel

Seien a, b, c, d *logische Variablen.*

$a + (b \cdot c)$ =

- $a + b \cdot c$
- $a + bc$

$\left(a \cdot \overline{b}\right) + \overline{(c \cdot d)}$ =

$a\overline{b} + \overline{cd}$

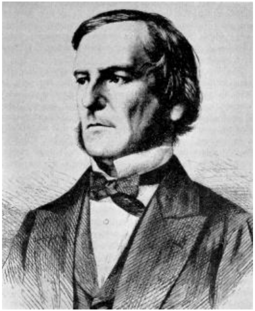
Achtung: $\overline{a} \cdot \overline{b} = \overline{a}\overline{b} \neq \overline{ab}$

Inhalt Digitale Systeme

Schaltalgebra

- mathematisches Konstrukt zur Berechnung und Vereinfachung von Digitalschaltungen
- Die Schaltalgebra kennt Variablen, Konstanten, Operationen und Axiome / Äquivalenzen.
  - Es gibt nur zwei mögliche Konstanten: 0 und 1.
  - Die Konstanten entsprechen den logischen Zuständen 0 und 1.
- Eine Variable kann entweder den Wert 0 oder 1 annehmen.
- Jede Größe, mit Wert 0 oder Wert 1, stellt eine Variable dar.
- Ausdrücke wie (a · b) können wie eine Variable behandelt werden, denn ihr Wert kann ebenfalls nur 0 oder 1 sein.
- Eine Variable der Schaltalgebra ist also eine binäre Größe.

George Boole



* 2. November 1815 in Lincoln, England
† 8. Dezember 1864 in Ballintemple, Irland
englischer Mathematiker (Autodidakt), Logiker und Philosoph

2.4 Boolesche Ausdrücke und Aussagen

Umformung Boolescher Ausdrücke

- Axiome: u. A. Regeln zur Vereinfachung und (Äquivalenz-) Umformung gegebener Ausdrücke
- Beschänkung auf boole'sche Ausdrücke (Konjunktion, Disjunktion und Negation)
- Alle anderen Operationen der Schaltalgebra lassen sich als boolesche Ausdrücke schreiben.
- Beweis durch Vergleich der Wertetabellen der Ausdrücke.

Definition

Eine Boole'sche Algebra A ist eine nichtleere Menge B mit den beiden zweistelligen Operation $a + b$, $a \cdot b$ *und* $\bar{a}$, *wobei B mindestens 2 verschiedene Elemente enthält und den folgenden Axiomen genügt:*

Umformung Boolescher Ausdrücke

Seien a, b, c **logische Variablen, dann gelten folgende Regeln:**

§1 Negation der Negation

$$\bar{\bar{a}} = a$$

§2 Kommutativgesetz

$$a \cdot b \cdot c = c \cdot a \cdot b$$

$$a + b + c = c + a + b$$

§3 Assoziativgesetz

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

$$a + (b + c) = (c + a) + b$$

§4 Distributivgesetz

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$a + (b \cdot c) = (a + b) \cdot (a + c)$$

§5 Idempotenzgesetz

$$a \cdot a = a$$

$$a + a = a$$

Umformung Boolescher Ausdrücke

§6 Komplementgesetz

$a \cdot \bar{a} = 0$ (Kontradiktion)

$a + \bar{a} = 1$ (Tautologie)

§7 0-1-Gesetz

$a \cdot 1 = a$

$a \cdot 0 = 0$

$a + 1 = 1$

$a + 0 = a$

§8 Absorptionsgesetze

$a \cdot (a + b) = a$

$a + (a \cdot b) = a$

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

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

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

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

Umformung Boolescher Ausdrücke

§9 De Morgan’sche Regeln

$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

$$\overline{a + b} = \bar{a} \cdot \bar{b}$$

§10 Resolutionsregel

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

§11 Konsensusregel

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

Mit Hilfe von Wertetabellen kann die Gültikeit der genannten Regeln bewiesen werden!

Umformung Boolescher Ausdrücke

Beweis

Exemplarisch soll der Beweis für das vorletzte Absorptionsgesetz durchgeführt werden.

$$\left(a \cdot \overline{b}\right) + b = a + b$$

a	b	$a \cdot \overline{b}$	$\left(a \cdot \overline{b}\right) + b$	a + b
0	*0*	*0*	*0*	*0*
0	*1*	*0*	*1*	*1*
1	*0*	*1*	*1*	*1*
1	*1*	*0*	*1*	*1*

Die Spalten für $\left(a \cdot \overline{b}\right) + b$ *und* $a + b$ *stimmen überein.*

Umformung Boolescher Ausdrücke: Hintergrund

Die angeführten Regeln erlauben es, komplizierte Boolesche Ausdrücke zu vereinfachen ... meistens.

Beispiel

$$
\begin{aligned}
& x + yz + (\bar{x} \cdot \overline{yz}) & \\
&= yz + x + (\bar{x} \cdot \overline{yz}) && \text{(Kommutativgesetz §2)} \\
&= yz + (x + \bar{x}) \cdot (x + \overline{yz}) && \text{(Distributivgesetz §4)} \\
&= yz + 1 \cdot (x + \overline{yz}) && \text{(Komplementgesetz §6)} \\
&= yz + (x + \overline{yz}) && \text{(0-1-Gesetz §7)} \\
&= yz + x + \overline{yz} && \text{(Vorrangsregeln (Folie 35))} \\
&= yz + \overline{yz} + x && \text{(Kommutativgesetz §2)} \\
&= 1 + x && \text{(Komplementgesetz §6)} \\
&= 1 && \text{(0-1-Gesetz §7)}
\end{aligned}
$$

Formal wahre, formal falsche Ausdrücke

- Ein häufig beschrittener Weg zur Vereinfachung von Ausdrücken ist die Suche nach “Teilausdrücken” ...
- ... insbesondere diejenigen, die immer logisch 0 oder immer 1 sind, um anschließend die 0-1-Gesetze anzuwenden.

Definition

Ein Ausdruck heißt formal wahrer Ausdruck, wenn er für jede Belegung den Wahrheitswert 1 liefert. Ein Ausdruck heißt formal falsch, wenn er für jede Belegung den Wahrheitswert 0 liefert.

Formal wahre, formal falsche Ausdrücke

Beispiel

Formal wahre Ausdrücke:

$a + \bar{a} = 1$

$ab + a\bar{b} + \bar{a}b + \bar{a}\bar{b} = 1$

$(a \rightarrow b) + (b \rightarrow a) = 1$

$(a \equiv b) + (\bar{a} \equiv b) = 1$

$(a \cdot b) + 1 = 1$

Formal falsche Ausdrücke:

$a \cdot \bar{a} = 0$

$\left(\overline{ab + a\bar{b} + \bar{a}b + \bar{a}\bar{b}}\right) = 0$

$a \oplus a = 0$

$a \cdot b \cdot c \cdot 0 = 0$

Einfache Sätze zur Umformungen von Ausdrücken

Der folgende Satz ist eine Verallgemeinerung der De Morgan'schen Regel.

Satz

Ein boolescher Ausdruck lässt sich negieren, indem man alle Operatoren + durch $\cdot$ und alle $\cdot$ durch + ersetzt und die Konstanten und Variablen negiert. Die Klammerungen bleiben dabei erhalten.

$$\overline{(a_1 \cdot a_2 \cdot a_3 \cdot \ldots \cdot a_n)} = \overline{a_1} + \overline{a_2} + \overline{a_3} + \ldots + \overline{a_n}$$

$$\overline{(a_1 + a_2 + a_3 + \cdots + a_n)} = \overline{a_1} \cdot \overline{a_2} \cdot \overline{a_3} \cdot \ldots \cdot \overline{a_n}$$

Einfache Sätze zur Umformungen von Ausdrücken

Der Beweis ist denkbar einfach ...

Beweis

$$\overline{(a_1 a_2 a_3 \dots a_n)} = \overline{\left((a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot a_{n-1}) a_n\right)} \;\; (wegen\, Vorranggr.)$$

$$= \overline{(a_1 a_2 \dots a_{n-1})} + \overline{a_n} \;\; (De\ Morgansche\ Regel)$$

$$= \overline{\left((a_1 \dots a_{n-2}) a_{n-1}\right)} + \overline{a_n}$$

$$= \overline{a_1} + \overline{a_2} + \overline{a_3} + \cdots + \overline{a_n}$$

2.4 Boolesche Ausdrücke und Aussagen

Einfache Sätze zur Umformungen von Ausdrücken

Satz

Eine Konjunktion $a_1 \cdot a_2 \cdot a_3 \cdot \ldots \cdot a_\mathrm{n}$ *ist für eine Belegung* $(a_1, a_2,\ldots, a_\mathrm{n})$ *falsch genau dann, wenn es (mindestens) ein i gibt mit* $a_i = 0, i \in 1{,}2, \ldots, \mathrm{n}$. *Eine Disjunktion* $a_1 + a_2 + \cdots + a_n$ *ist für eine Belegung* $(a_1, a_2,\ldots, a_\mathrm{n})$ *richtig genau dann, wenn es (mindestens) ein i gibt mit* $a_i = 1, i \in 1{,}2, \ldots, \mathrm{n}$.

Satz

Eine Konjunktion $a_1 \cdot a_2 \cdot \ldots \cdot a_\mathrm{n}$*ist genau dann wahr, wenn alle Variablen* a_i *mit 1 belegt sind. Eine Disjunktion* $a_1 + a_2 + \cdots + a_n$ *ist genau dann falsch, wenn alle Variablen* a_i *mit 0 belegt sind.*

Hintergrund: Diese Aussagen werden später bei der Behandlung der Normalformen benötigt.

Inhalt Digitale Systeme

1. Motivation und Einführung
2. Logik und Boolesche Algebra
   - ✓ Digital vs. Analog
   - ✓ Grundbegriffe der Logik
   - ✓ Logische Operationen
   - ✓ Boolesche Ausdrücke und Aussagen
   - ❑ Logische Funktionen, Bündel
3. Gatter und CMOS-Technik
4. Zahlendarstellung und Codes
5. Schaltnetze und Normalformen
6. Optimierung von Schaltnetzen
7. Standard-Schaltnetze
8. Speicherelemente und programmierbare Logik
9. Synchrone Schaltwerke
10. Register-Transfer-Entwurf und Mikroprogrammierung
11. Anwendungen Digitaler Systeme

2.5 Logische Funktionen, Bündel

Begriff der logischen Funktionen

- Eine logische Funktion beschreibt eine Zuordnung zwischen einer Belegung der Variablen und den Wahrheitswerten 0 und 1

- Das Ergebnis ist eine Funktion der (unabhängigen) Variablen

- $y := f(a_1, a_2, \ldots, a_n)$

Definition

n-stellige logische Funktion

Sei $D \subset \{0{,}1\}^n, n \in \mathbb{N}$ *und* $(a_1, a_2, \ldots, a_n) \in D$ *eine Belegung.* $\{0\ ,1\ \}^n$ *ist Menge aller* n-Tupel $(a_1, a_2, \ldots, a_n)$ *mit* $a_i \in \{0{,}1\}$ *für alle* $i = 1{,}2, \ldots, n$

Eine Zuordnungsvorschrift $f\colon D \to \{0{,}1\}$ *mit* $(a_1, \ldots, a_n) \mapsto f(a_1, a_2, \ldots, a_n)$ *heißt n-stellige logische Funktion oder logische Funktion genau dann, wenn f eindeutig ist, d.h. jeder Belegung* $(a_1, a_2, \ldots, a_n) \in D$ *wird genau ein Wert* $(a_1, a_2, \ldots, a_n) \in \{0{,}1\}$ *zugeordnet.*

$f(a_1, a_2, \ldots, a_n)$ *heißt Funktionswert, D heißt Definitionsbereich, die Elemente aus D heißen Argumente der Funktion. Ist* $D = \{0{,}1\}^n$*, so heißt* f *vollständig definiert. Ist* $D \subset \{0{,}1\}^n$*, so heißt* f *partiell definiert.*

2.5 Logische Funktionen, Bündel

Vollstängig definierte logische Funktionen

- Eine logische Funktion kann durch eine Wertetabelle oder durch einen logischen Ausdruck angegeben werden.
- Bei vollständig definierten Funktionen muss der Funktionswert unter jeder Belegung eindeutig definiert sein (entweder 0 oder 1).

Beispiel

x_1	x_2	$g(x_1, x_2)$
0	*0*	*0*
0	*1*	*1*
1	*0*	*0*
1	*1*	*0*

$$f(a,b,c) = \overline{a+b} \rightarrow c$$

Partiell definierte logische Funktionen

- Eine logische Funktion muss nicht für jede Belegung ihrer Variablen definiert sein ($D \subset \{0,1\}^n$)

- Der undefinierte Funktionswert wird als "Don't Care" bezeichnet (Symbol: - oder *).

- partiell definierte logische Funktionen interessant für Optimierung

Beispiel

x_1	x_2	$g(x_1, x_2)$
0	*0*	-
0	*1*	*1*
1	*0*	*0*
1	*1*	-

Hinweis: Logische Ausdrücke sind nie partiell definierte Funktionen!

2.5 Logische Funktionen, Bündel

Funktionsgleichung

- Eine Funktion kann auch durch eine abhängige Variable definiert werden.
- Diese kann selbst in einer Funktion als Variable auftreten.
- Dadurch lassen sich logische Funktionen verschachteln oder verknüpfen.

Beispiel

$$y = f(a, b)$$

$$\begin{aligned} z &= g(a, c, y) \\ &= g(a, c, f(a, b)) \end{aligned}$$

Hinweis: Man spricht hierbei auch von Funktionsgleichungen. Diese werden im Zusammenhang mit der Analyse und Synthese von größeren Schaltnetzen zur besseren Übersicht benutzt (z.B. beim Addierer).

Vollständige Systeme

Eine logische Funktion kann durch verschiedene zueinander äquivalente Ausdrücke angegeben werden.

Satz

Äquivalenzen zu Grundverknüpfungen

Seien x_1, x_2 *logische Variable, dann gilt:*

$$f(x_1, x_2) = x_1 \cdot x_2 = \overline{(\overline{x_1} + \overline{x_2})}$$

$$f(x_1, x_2) = x_1 \oplus x_2 = \overline{x_1} x_2 + x_1 \overline{x_2} = (\overline{x_1 + \overline{x_2}}) + (\overline{\overline{x_1} + x_2})$$

$$f(x_1, x_2) = x_1 + x_2 = \overline{(\overline{x_1} \cdot \overline{x_2})}$$

$$f(x_1, x_2) = \overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2}$$

$$f(x_1, x_2) = x_1 \equiv x_2 = \overline{x_1}\,\overline{x_2} + x_1 x_2 = (\overline{x_1 + x_2}) + (\overline{\overline{x_1} + \overline{x_2}})$$

$$f(x_1, x_2) = x_1 \rightarrow x_2 = (\overline{x_1 + \overline{x_2}}) = \overline{x_1} + x_2$$

$$f(x_1, x_2) = \overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2}$$

Vollständige Systeme

- Beweis durch Wertetabellen und die logischen Gesetze
- Gleichungen zeigen, dass sich alle Grundverknüpfungen durch $\cdot$, $+$, $\overline{}$
- sogar $+$ und $\overline{}$ genügen aus!

Satz

Boolescher Ausdruck

Seien $a_1, a_2, \dots, a_n$ *logische Variable. Jede vollständig definierte logische Funktion* $g(a_1, a_2, \dots, a_n)$ *lässt sich als logischer Ausdruck, der nur aus den Grundverknüpfungen* $\cdot$, $+$, $\overline{}$ *zusammengesetzt ist, d.h. als boolescher Ausdruck, angeben.*

2.5 Logische Funktionen, Bündel

Vollständige Systeme

Definition

Eine Menge logischer Operationen $(O_1, O_2, \ldots, O_k)$ *heißt vollständiges System, wenn sich jede vollständig definierte logische Funktion unter ausschließlicher Verwendung dieser Operationen bilden lässt.*

Beweis

Seien a, b *logische Variablen.*

$a + b$ *bzw.* $a \cdot b$ *ssen sich mit* $\cdot$ und $\overline{\ }$ *bzw.* $+$ *und* $\overline{\ }$ *angeben.*

$(\cdot, \overline{\ })$: $a + b = \overline{\overline{a} \cdot \overline{b}}$ *(De Morgansche Regel)*

$(+, \overline{\ })$: $a \cdot b = \overline{\overline{a} + \overline{b}}$ *(De Morgansche Regel)*

Satz

<u>NOR</u> und <u>NAND</u> bilden jeweils ein vollständiges System.

Funktionsbündel

- In der Praxis sind häufig Funktionsbündel zu betrachten.
- Es handelt sich dabei um eine (endliche) Menge logischer Funktionen, die das gleiche Argument haben.

Definition

Eine Zusammenfassung σ logischer Funktionen $f_1, f_2, \ldots, f_m$*, die das gleiche Argument* $(a_1, \ldots, a_n)$ *haben* ($m, n \in \mathbb{N}$)*, heißt ein* ***Funktions-bündel*** *oder eine Vektorfunktion.*

Schreibweise:

$$\sigma(a_1, a_2, \ldots, a_n) = (y_1, y_2, \ldots, y_m)$$

mit

$$y_1 = f_1(a_1, \ldots, a_n)$$
$$y_2 = f_2(a_1, \ldots, a_n)$$
$$\ldots$$
$$y_m = f_m(a_1, \ldots, a_n)$$

Funktionsbündel

Beispiel

Funktionsbündel werden z.B. bei der Umcodierung eines Binärcodes in einen anderen, wie es z.B. bei der Umformung des BCD-Codes in den 1-aus-10-Code der Fall ist, benötigt. Zur Beschreibung der Ausgänge werden zehn verschiedene Funktionen benötigt, die vier gemeinsame Variablen an den Eingängen haben. z. B.

$$y_1 = x_1 x_2 x_3 + x_1 x_4$$

$$y_2 = x_1 \overline{x_4} + x_1 x_2 x_3$$

2.5 Logische Funktionen, Bündel

Darstellung von Funktionsbündeln

- Darstellung von Funktionsbündeln durch eine Anzahl m verschiedener logischer Ausdrücke ...
- ... oder durch eine Wertetabelle

Beispiel

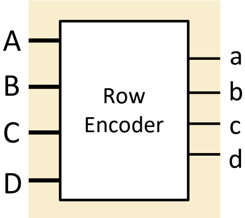
Funktionsbündel $\sigma(a,b,c) = (x, y, z, t)$

a	b	c	x	y	z	t
0	0	0	1	1	1	0
0	0	1	1	1	1	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
1	0	0	0	0	0	1
1	0	1	1	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	0	1

Anwendung von Funktionsbündeln

- **Minimierung:** Zusammenfassen mehrerer Funktionen zur Minimierung des Hardwareaufwands
- **gemeinsame Teilausdrücke:** Gibt es in mehreren der beteiligten Funktionen (logischen Ausdrücken) gemeinsame Teile, kann dieser Teil von mehreren Funktionen gleichzeitig benutzt werden.
- Einsparung von Hardware und Verlustleistung bei der technischen Realisierung
- Bisher kein allgemeiner Algorithmus zur effizienten Berechnung optimaler Lösungen von mehrstufigen Schaltnetzen.

Zentrale Fragen Kapitel 2



- Was heißt „digital“?
- Was ist eine logische Operation?
- Wie rechnet man mit Boolescher Algebra?
- Gibt es mehrstellige logische Operationen?

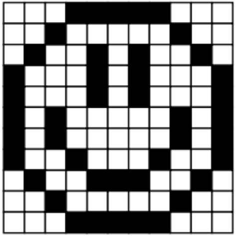
z.B. als logische Funktionen

$$a = D + \bar{A}\bar{C} + AB$$
$$b = \bar{C} + AB + \bar{A}\bar{B}$$
$$c = \ \ldots$$

… oder Wahrheitstabellen

Input				Output			
D	C	B	A	a	b	c	…
0	0	0	0	1	1	1	…
0	0	0	1	0	1	1	…
0	0	1	0	1	1	0	…
0	0	1	1	1	1	1	…

Register-Transfer
Schaltnetze & Schaltwerke
Gatter
Transistoren
Halbleiter



0	0	0	1	1	1	1	1	0	0	0
0	0	1	0	0	0	0	0	1	0	0
0	1	0	0	1	0	1	0	0	1	0
1	0	0	0	1	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0	1
1	0	1	0	0	0	0	0	1	0	1
1	0	1	0	0	0	0	0	1	0	1
1	0	0	1	0	0	0	1	0	0	1
0	1	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0
0	0	0	1	1	1	1	1	0	0	0

- ✓ Was heißt „digital“?
- ❑ Was ist eine logische Operation?
- ❑ Wie rechnet man mit Boolescher Algebra?
- ❑ Gibt es mehrstellige logische Operationen?

Zentrale Fragen Kapitel 2

Register-Transfer
Schaltnetze & Schaltwerke
Gatter
Transistoren
Halbleiter

a	b		$a \wedge b$		a		b	$a \not\equiv b$	$a \vee b$	$\overline{a \vee b}$	$a \equiv b$	$\bar{b}$		$\bar{a}$	$a \rightarrow b$	$\overline{a \wedge b}$	
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- ✓ Was heißt „digital"?
- ✓ Was ist eine logische Operation?
- ❑ Wie rechnet man mit Boolescher Algebra?
- ❑ Gibt es mehrstellige logische Operationen?

Zentrale Fragen Kapitel 2

Umformung Boolescher Ausdrücke
Seien a, b, c **logische Variablen, dann gelten folgende Regeln:**

§1 Negation der Negation

$\bar{\bar{a}} = a$

§2 Kommutativgesetz

$a \cdot b \cdot c = c \cdot a \cdot b$

$a + b + c = c + a + b$

§3 Assoziativgesetz

$a \cdot (b \cdot c) = (a \cdot b) \cdot c$

$a + (b + c) = (c + a) + b$

§4 Distributivgesetz

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

$a + (b \cdot c) = (a + b) \cdot (a + c)$

§5 Idempotenzgesetz

$a \cdot a = a$

$a + a = a$

§6 Komplementgesetz

$a \cdot \bar{a} = 0$ (Kontradiktion)

$a + \bar{a} = 1$ (Tautologie)

§7 0-1-Gesetz

$a \cdot 1 = a$

$a \cdot 0 = 0$

$a + 1 = 1$

$a + 0 = a$

§8 Absorptionsgesetze

$a \cdot (a + b) = a$

$a + (a \cdot b) = a$

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

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

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

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

§9 De Morgan'sche Regeln

$\overline{a \cdot b} = \bar{a} + \bar{b}$

$\overline{a + b} = \bar{a} \cdot \bar{b}$

10§ Resolutionsregel

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

11§ Konsensusregel

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

- ✓ Was heißt „digital"?
- ✓ Was ist eine logische Operation?
- ✓ Wie rechnet man mit Boolescher Algebra?
- ❑ Gibt es mehrstellige logische Operationen?

Beispiel

Funktionsbündel $\sigma(a, b, c) = (x, y, z, t)$

a	b	c	x	y	z	t
0	0	0	1	1	1	0
0	0	1	1	1	1	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
1	0	0	0	0	0	1
1	0	1	1	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	0	1

- ✓ Was heißt „digital“?
- ✓ Was ist eine logische Operation?
- ✓ Wie rechnet man mit Boolescher Algebra?
- ✓ Gibt es mehrstellige logische Operationen?

Danke für Ihre Aufmerksamkeit!

http://www.magicbluesmoke.org