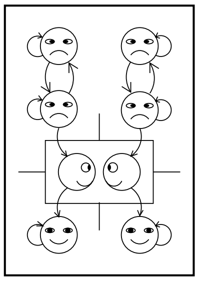
Technische Universität Ilmenau

Fakultät für Informatik und Automatisierung
Institut für Technische Informatik und Ingenieurinformatik
Fachgebiet Integrierte Hard- und Softwaresysteme



Arbeitsblätter

zur Lehrveranstaltung

Technische Informatik – Teil RO

(Studiengänge EIT, FZT, LA, MB, MT, MTR, OTR, WSW)

(Ausgabe Oktober 2009)

Dr.-Ing. Heinz-Dietrich Wuttke

Dr.-Ing. Karsten Henke

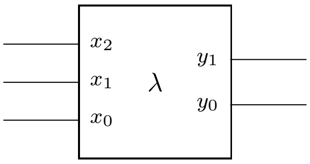
Inhaltsübersicht

Funktionsbeschreibung digitaler Schaltungen (kombinatorisch)

Digitale Schaltung

$X = \{X_0, \cdots, X_i, \cdots, X_{2^n-1}\}$ $\qquad\qquad$ $Y = \{Y_0, \cdots, Y_t, \cdots, Y_{2^m-1}\}$

$$\begin{matrix} 0 & \cdots & 0 & \cdots & 1 \\ 0 & \cdots & 0 & \cdots & 1 \\ 0 & \cdots & 1 & \cdots & 1 \end{matrix}$$



$$\begin{matrix} 0 & \cdots & 0 & \cdots & 1 \\ 0 & \cdots & 1 & \cdots & 1 \end{matrix}$$

mit:

- X – Menge der Eingangsbelegungen X_i des Eingangsvektors $x = [x_{n-1}, \ldots, x_r, \ldots, x_0]$
- Y – Menge der Ausgangsbelegungen Y_t des Ausgangsvektors $y = [y_{m-1}, \ldots, y_k, \ldots, y_0]$
- λ – Abbildungsvorschrift (Funktion der Schaltung):

$\lambda : X \Rightarrow Y$ bzw. $\lambda(X) = Y$ (determiniert)

$\lambda : X \Rightarrow \mathcal{P}(Y)\backslash\emptyset$ bzw. $\lambda(X) = \mathcal{P}(Y)\backslash\emptyset$ (nichtdeterminiert)

Definitionen

Ein – /Ausgangsvektor $\quad x = [x_{n-1}, \ldots, x_r, \ldots, x_0]$ bzw. $y = [y_{m-1}, \ldots, y_k, \ldots, y_0]$ aus binären Variablen x_r bzw. y_k mit n bzw. m als Stelligkeit von x bzw. y

Eingangsbelegung $\quad X_i(x) \Rightarrow \{0,1\}^n$ geordnete Menge von n Bits; $n = |x|$

$X_i = [X_i(x_{n-1}), \ldots, X_i(x_r), \ldots, X_i(x_0)]$

Bit der Eingangsbelegung $\quad X_i(x_r) \in \{0,1\}$ Wert der Eingangsvariablen x_r in der Belegung X_i

Ausgangsbelegung $\quad Y_t(y) \Rightarrow \{0,1\}^m$ (determiniert)

$Y_t(y) \Rightarrow \{0,1,*,H(g)\}^m$ (nichtdeterminiert siehe Beispiel Seite 3)

$$\begin{aligned} Y_t &= [Y_t(y_{m-1}), \ \ldots \ , Y_t(y_k), \ \ldots, Y_t(y_0)] \\ &= [\lambda_{m-1}(X_i), \ldots, \lambda_k(X_i), \ldots, \lambda_0(X_i)] = \lambda(X_i) \end{aligned}$$

Bit der Ausgangsbelegung $\quad Y_t(y_k) \in \{0,1\}$ (determiniert)

$Y_t(y_k) \in \{0,1,*,H(g)\}$ (nichtdeterminiert)

Belegungsmenge $\quad X = \{X_i \mid 0 \leq i \leq 2^n - 1\}$ $\qquad$ $Y = \{Y_t \mid 0 \leq t \leq 2^m - 1\}$

Belegungsindizes $\quad i = \sum_{r=0}^{n-1} X_i(x_r) \cdot 2^r$ $\qquad$ $t = \sum_{k=0}^{m-1} Y_t(y_k) \cdot 2^k$

Indexmenge $\quad$ Menge der Indizes der Eingangsbelegungen

$M = \{i \mid 0 \leq i \leq 2^n - 1\}$; $\quad |M| = |X|$

Wertetabelle

$\lambda : X \Rightarrow Y$ — Belegung Y des Ausgangsvektors y als Funktion $\lambda(X)$ der Belegung X des Eingangsvektors x.

BI	x_{n-1}	$\cdots$	x_r	$\cdots$	x_0	y_{m-1}	$\cdots$	y_k	$\cdots$	y_0	BI
0	0	$\cdots$	0	$\cdots$	0						
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
i	$X_i(x_{n-1})$	$\cdots$	$X_i(x_r)$	$\cdots$	$X_i(x_0)$	$\lambda_{m-1}(X_i)$	$\cdots$	$\lambda_k(X_i)$	$\cdots$	$\lambda_0(X_i)$	t
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
$2^n - 1$	1	$\cdots$	1	$\cdots$	1						

Beispiel: Funktion mit 3 Eingangs- und 2 Ausgangsvariablen

Ein-/Ausgangsvektor: $x = [x_2, x_1, x_0]$ mit $|x| = 3$ bzw. $y = [y_1, y_0]$ mit $|y| = 2$

Eingangsbelegung: $X_5(x) = [1, 0, 1]$ als geordnete Menge von $n = |x| = 3$ Bits mit dem Belegungsindex $i = 5 = \mathbf{1} \cdot 2^2 + \mathbf{0} \cdot 2^1 + \mathbf{1} \cdot 2^0)$

Bit der Eingangsbelegung: $X_5(x_0) = 1$ das "0"-te Bit der Eingangsbelegung X_5

Ausgangsbelegung: $\lambda(X_5) = [\lambda_1(X_5), \lambda_0(X_5)] = [Y_2(y_1), Y_2(y_0)] = [1, 0] = Y_2$

Wertetabelle:

$y = [y_1, y_0]$

$x = [x_2, x_1, x_0]$

i	$\boldsymbol{x_2}$	$\boldsymbol{x_1}$	$\boldsymbol{x_0}$	$\boldsymbol{y_1}$	$\boldsymbol{y_0}$	t
0	0	0	0	1	1	3
1	0	0	1	0	1	1
2	0	1	0	0	0	0
3	0	1	1	1	0	2
4	1	0	0	1	1	3
5	1	0	1	1	0	2
6	1	1	0	1	1	3
7	1	1	1	0	1	1

$X_2 = [0, 1, 0]$

$X_5(x_1) = 0$

$X_6(x_2) = 1$

$\lambda_1(X_1) = Y_1(y_1) = 0$

$\lambda_0(X_3) = Y_2(y_0) = 0$

$\lambda(X_5) = Y_2 = [1, 0]$

Schaltalgebraische Ausdrücke

schaltalgebraischer Ausdruck

$h_i(x), h_j(x) \in H$ sind elementare oder zusammengesetzte schaltalgebraische Ausdrücke und sind folgendermaßen induktiv definiert:

1. Konstante 0 und 1 sind schaltalgebraische Ausdrücke;
2. binäre Variable x_r eines n-stelligen Vektors x sind schaltalgebraische Ausdrücke;
3. wenn $h_i(x)$ und $h_j(x)$ Ausdrücke sind, so auch:

$$\begin{array}{ll} \overline{h_i(x)}, & (h_i(x) \rightarrow h_j(x)), \\ (h_i(x) \wedge h_j(x)), & (h_i(x) \sim h_j(x)), \\ (h_i(x) \vee h_j(x)), & (h_i(x) \not\sim h_j(x)) \end{array}$$

4. andere Zeichenketten sind keine schaltalgebraischen Ausdrücke.

Wertbestimmung für schaltalgebraische Ausdrücke

Wertfunktion

Die Wertfunktion $\mathcal{W}$ ordnet einem *schaltalgebraischen Ausdruck* $h_j \in H$ bei einer *Belegung* $X_i \in X$ einen Wert aus $\{0, 1\}$ zu.

$\mathcal{W} : H \times X \Rightarrow \{0, 1\}$ z.B. $\mathcal{W}(h_j(x), X_i) = 0$

X − Partitionierung

Jeder schaltalgebraische Ausdruck $h_j(x)$ teilt die Belegungsmenge X des Vektors x disjunkt in zwei Teilmengen X^1 und X^0, wobei gilt:

- $X = X^1 \cup X^0$; $X^1 \cap X^0 = \emptyset$; $X^1 = \overline{X^0}$
- $\forall X_i(X_i \in X^1 \leftrightarrow \mathcal{W}(h_j(x), X_i) = 1)$
- $\forall X_i(X_i \in X^0 \leftrightarrow \mathcal{W}(h_j(x), X_i) = 0)$

Wertbestimmung

$\forall X_i(\mathcal{W}(h_j(x), X_i) = 1 \leftrightarrow p_j(X_i))$

sprich: *Für alle Belegungen X_i gilt: Der Wert eines schaltalgebraischen Ausdrucks $h_j(x)$ bei der Belegung X_i ist 1, genau dann, wenn die von X_i abhängige Aussage $p_j(X_i)$ wahr ist und 0, falls $p_j(X_i)$ falsch ist.*

	$h_j(x)$	$p_j(X_i)$
(a)	0	f
(b)	1	w
(c)	x_k	$X_i(x_k) = 1$
(d)	$\overline{x_k}$	$X_i(x_k) = 0$
(e)	$\overline{h_k(x)}$	$\overline{\mathcal{W}(h_k(x), X_i) = 1}$
(f)	$h_k(x) \wedge h_l(x)$	$(\mathcal{W}(h_k(x), X_i) = 1) \wedge (\mathcal{W}(h_l(x), X_i) = 1)$
(g)	$h_k(x) \vee h_l(x)$	$(\mathcal{W}(h_k(x), X_i) = 1) \vee (\mathcal{W}(h_l(x), X_i) = 1)$

z.B. Zeile (d): $\forall X_i\big(\mathcal{W}(\overline{x_k}, X_i) = 1 \leftrightarrow X_i(x_k) = 0\big)$

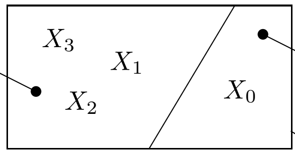
verbal: *Der Wert des Ausdrucks $\overline{x_k}$ bei der Belegung X_i ist gleich 1, genau dann, wenn Bit k der Belegung X_i gleich 0 ist.*

Beispiel

$h_i = x_0 \vee x_1 \quad x = [x_1, x_0] \quad X = \{X_3, X_2, X_1, X_0\}$

X-Partitionierung

$h_i = x_0 \vee x_1 \;\hat{=}\; X^1$



$\overline{h_i} = \overline{x_0 \vee x_1} = \overline{x_0} \wedge \overline{x_1} \;\hat{=}\; \overline{X^1} = X^0$

$X = X^1 \cup X^0$

Wertbestimmung

$\mathcal{W}\big((x_0 \vee x_1), X_0\big) = \mathcal{W}(x_0, X_0) \vee \mathcal{W}(x_1, X_0) = X_0(x_0) \vee X_0(x_1) = 0 \vee 0 = 0$

$\mathcal{W}\big((x_0 \vee x_1), X_2\big) = \mathcal{W}(x_0, X_2) \vee \mathcal{W}(x_1, X_2) = X_2(x_0) \vee X_2(x_1) = 0 \vee 1 = 1$

$\mathcal{W}\big((x_0 \vee \overline{x_1}), X_2\big) = \mathcal{W}(x_0, X_2) \vee \overline{\mathcal{W}(x_1, X_2)} = X_2(x_0) \vee \overline{X_2(x_1)} = 0 \vee \overline{1} = 0$

BOOLEsche Ausdrucksalgebra (BAA)

$BAA = [K, \vee, \wedge, \bar{\ }, 0, 1]$ mit:

- K als Menge von Repräsentanten h_i je einer unendlichen Menge H^i werteverlaufsgleicher Ausdrücke a_i, wobei gilt:
  - $a_i \underset{o}{=} h_i \leftrightarrow h_i, a_i \in H^i \quad |H^i| = \infty$, $|X| = 2^n$, $0 \leq i \leq |K| - 1$
  - $|K| = |P(X)| = 2^{2^n}$ mit n = Anzahl der x-Variablen

  z.B. für $n = 2$: $|X| = 2^2 = 4$; $|K| = 2^{2^2} = 16$ (siehe auch Seite 13)

  $K = \{h_0, h_1, \ldots, h_{15}\}$

  $H^9 = \{x_1 x_0 \vee \bar{x}_1 \bar{x}_0, (\bar{x}_1 \vee x_0)(x_1 \vee \bar{x}_0), x_1 \sim x_0, \ldots\}$

  $h_9 = x_1 x_0 \vee \bar{x}_1 \bar{x}_0$ als Repräsentant aus H^9 in DNF
- $\vee, \wedge, \bar{\ }$ als Operationen
- 0 als neutrales Element der Disjunktion ($\vee$)
- 1 als neutrales Element der Konjunktion ($\wedge$)

Axiome und Regeln der BOOLEschen Ausdrucksalgebra

Kommutativität

$h_i \vee h_j \underset{o}{=} h_j \vee h_i$

$h_i \wedge h_j \underset{o}{=} h_j \wedge h_i$

Assoziativität

$h_i \vee (h_j \vee h_k) \underset{o}{=} (h_i \vee h_j) \vee h_k \underset{o}{=} h_i \vee h_j \vee h_k$

$h_i \wedge (h_j \wedge h_k) \underset{o}{=} (h_i \wedge h_j) \wedge h_k \underset{o}{=} h_i \wedge h_j \wedge h_k$

Distributivität

$h_i \vee (h_j \wedge h_k) \underset{o}{=} (h_i \vee h_j) \wedge (h_i \vee h_k)$

$h_i \wedge (h_j \vee h_k) \underset{o}{=} (h_i \wedge h_j) \vee (h_i \wedge h_k)$

Idempotenz

$h_i \vee h_i \underset{o}{=} h_i$

$h_i \wedge h_i \underset{o}{=} h_i$

Adjunktivität

$h_i \wedge (h_i \vee h_j) \underset{o}{=} h_i$

$h_i \vee (h_i \wedge h_j) \underset{o}{=} h_i$

Negation

$h_i \vee \bar{h}_i \underset{o}{=} 1$

$h_i \wedge \bar{h}_i \underset{o}{=} 0$

$\bar{\bar{h}}_i \underset{o}{=} h_i$

$\bar{0} \underset{o}{=} 1$

$\bar{1} \underset{o}{=} 0$

*Disjunktions*regel	$h_i \vee 0 \underset{\text{o}}{=} h_i$
	$h_i \vee 1 \underset{\text{o}}{=} 1$
*Konjunktions*regel	$h_i \wedge 0 \underset{\text{o}}{=} 0$
	$h_i \wedge 1 \underset{\text{o}}{=} h_i$
*deMORGAN*sche Regel	$\overline{h_i \vee h_j} \underset{\text{o}}{=} \bar{h}_i \wedge \bar{h}_j$
	$\overline{h_i \wedge h_j} \underset{\text{o}}{=} \bar{h}_i \vee \bar{h}_j$
Implikationsregel	$h_i \rightarrow h_j \underset{\text{o}}{=} \bar{h}_i \vee h_j$
Äquivalenzregel	$h_i \sim h_j \underset{\text{o}}{=} h_i h_j \vee \bar{h}_i \bar{h}_j$
Antivalenzregel	$h_i \not\sim h_j \underset{\text{o}}{=} \overline{h_i \sim h_j} \underset{\text{o}}{=} h_i \bar{h}_j \vee \bar{h}_i h_j$

Wichtige Kürzungsregeln

1. $h_i h_j \vee \bar{h}_i h_j \underset{\text{o}}{=} (h_i \vee h_j)(\bar{h}_i \vee h_j) \underset{\text{o}}{=} h_j$
2. $h_i \vee h_i h_j \underset{\text{o}}{=} h_i(h_i \vee h_j) \underset{\text{o}}{=} h_i$
3. $h_i \vee \bar{h}_i h_j \underset{\text{o}}{=} h_i \vee h_j$
4. $h_i(\bar{h}_i \vee h_j) \underset{\text{o}}{=} h_i h_j$
5. $h_i h_j \vee h_i \bar{h}_k \vee h_j h_k \underset{\text{o}}{=} h_i \bar{h}_k \vee h_j h_k$
6. $(h_i \vee h_j)(h_i \vee \bar{h}_k)(h_j \vee h_k) \underset{\text{o}}{=} (h_i \vee \bar{h}_k)(h_j \vee h_k)$

Elementarkonjunktion und -disjunktion

Elementarkonjunktion $\quad k_i(x) ::= \bigwedge_{r=0}^{n-1}(X_i(x_r) \sim x_r)$

Beispiel: Ermittlung von $k_3(x) = \overline{x_2}x_1x_0$

BI	x_2	x_1	x_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
⋮	⋮	⋮	⋮

$$\begin{aligned} k_3(x) &::= (X_3(x_2) \sim x_2) \wedge (X_3(x_1) \sim x_1) \wedge (X_3(x_0) \sim x_0) \\ &::= (0 \sim x_2)(1 \sim x_1)(1 \sim x_0) \\ &::= (0x_2 \vee 1\overline{x_2})(1x_1 \vee 0\overline{x_1})(1x_0 \vee 0\overline{x_0}) \\ &::= (0 \vee \overline{x_2})(x_1 \vee 0)(x_0 \vee 0) \\ &::= \overline{x_2}x_1x_0 \end{aligned}$$

Elementardisjunktion $\quad d_i(x) ::= \bigvee_{r=0}^{n-1}(X_i(x_r) \not\sim x_r)$

Ermittlung **expliziter BOOLEscher Gleichungen in Normalform**
für je eine Ausgangsvariable $y_k \in y$ entsprechend folgender Definitionen:

KDNF

Kanonisch
Disjunktive
Normalform

$$y_k = \bigvee_{i=0}^{2^n-1} k_i(x) \wedge \lambda_k(X_i)$$

$$h^*(x) \underset{\circ}{=} \bigvee_{i \in M_k} k_i(x) \qquad \text{mit } \forall i(i \in M_k \leftrightarrow \lambda_k(X_i) = *)$$

KKNF

Kanonisch
Konjunktive
Normalform

$$y_k = \bigwedge_{i=0}^{2^n-1} (d_i(x) \vee \lambda_k(X_i))$$

$$\overline{h^*(x)} \underset{\circ}{=} \bigwedge_{i \in M_k} d_i(x) \qquad \text{mit } \forall i(i \in M_k \leftrightarrow \lambda_k(X_i) = *)$$

KNANF

Kanonische
NAND-
Normalform

$$KDNF \xleftrightarrow[\text{und deMorgan}]{\text{doppelteNegation}} KNANF$$

$$y_k = \overline{\bigwedge_{i=0}^{2^n-1} \overline{k_i(x) \wedge \lambda_k(X_i)}}$$

KNONF

Kanonische
NOR-
Normalform

$$KKNF \xleftrightarrow[\text{und deMorgan}]{\text{doppelteNegation}} KNONF$$

$$y_k = \overline{\bigvee_{i=0}^{2^n-1} \overline{d_i(x) \vee \lambda_k(X_i)}}$$

Ausgangspunkt: Wertetabelle

Idee:

- Grafische Aufstellung der Wertetabelle so, daß benachbarte Belegungen auch in der Tabelle benachbart sind.
  - Zwei Belegungen X_i und X_j heißen **benachbart**, wenn sie sich in genau einem Bit an der r-ten Stelle unterscheiden, d.h. es gilt:

$$X_i(x_s) = \begin{cases} \overline{X_j(x_s)} & \text{falls} \quad s = r \\ X_j(x_s) & \text{sonst} \end{cases} \qquad (1)$$

  - Elementarkonjunktionen benachbarter Belegungen sind in der Variablen x_r kürzbar zu Fundamental-Konjunktionen entsprechend folgender Kürzungsregeln:

$$h_i(x) \underset{\circ}{=} h_i(x)h_j(x) \vee h_i(x)\overline{h_j(x)}$$
$$h_i(x) \underset{\circ}{=} x_r h_i(x) \vee \bar{x_r} h_i(x)$$

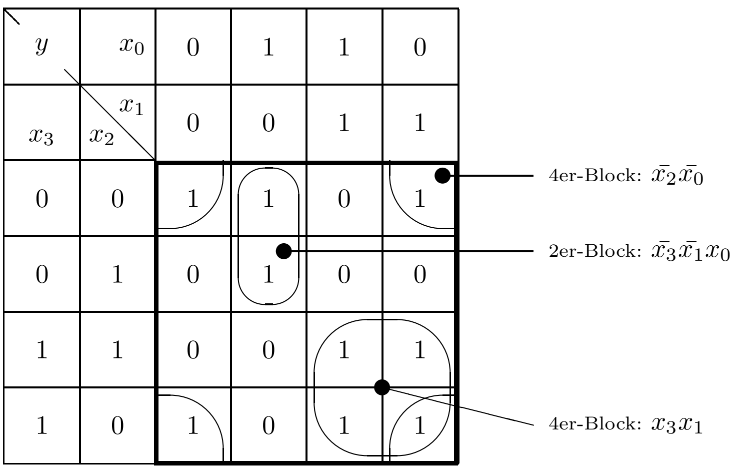
- Grafische Gruppenbildung benachbarter Belegungen

Verfahren:

- zwei im Karnaugh-Plan benachbarte Felder erfüllen die Nachbarschaftsbeziehung (1)
- linker und rechter sowie oberer und unterer Rand des Karnaugh-Planes sind benachbart
- $\Rightarrow$ Minimierung durch Bilden von 2er-, 4er-, 8er-, … Blöcken untereinander benachbarter Felder
- die Variablen, deren Wert innerhalb eines Blockes konstant ist, bilden den (diese Belegungen repräsentierenden) Minimalausdruck

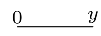
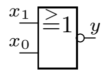
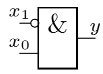
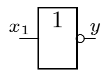
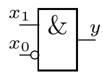
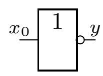
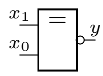
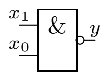
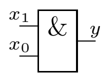
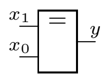
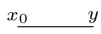
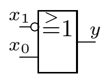
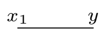
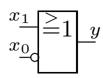
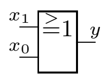
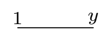
Beispiel

x_3	x_2	x_1	x_0	y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1



$$y_{min} = \bar{x_2}\bar{x_0} \vee x_3x_2x_1x_0 \vee \bar{x_3}x_2\bar{x_1}x_0$$

Beispiele für Repräsentanten h_i der Mengen H^i für Ausdrücke mit 2 Variablen x_0, x_1 in unterschiedlichen Normalformen

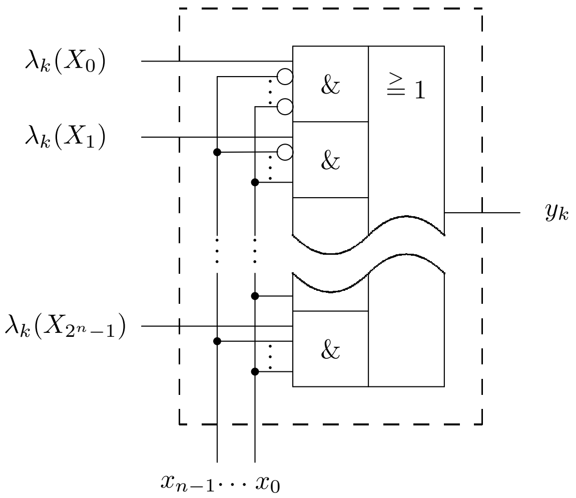
x_1 / x_0	1 1 0 0 / 1 0 1 0	Funktionsname	DNF	KNF	weitere NF	Schalt-symbol
y_0	0 0 0 0	Null	0	0	0	
y_1	0 0 0 1	NOR (not or)	$\bar{x}_1\bar{x}_0$	$\bar{x}_1\bar{x}_0$	$\overline{x_1 \vee x_0}$	
y_2	0 0 1 0	Inhibition von x_0 auf x_1	$\bar{x}_1 x_0$	$\bar{x}_1 x_0$	$\overline{x_0 \rightarrow x_1}$	
y_3	0 0 1 1	NOT (Negation von x_1)	$\bar{x}_1$	$\bar{x}_1$	$\bar{x}_1$	
y_4	0 1 0 0	Inhibition von x_1 auf x_0	$x_1\bar{x}_0$	$x_1\bar{x}_0$	$\overline{x_1 \rightarrow x_0}$	
y_5	0 1 0 1	NOT (Negation von x_0)	$\bar{x}_0$	$\bar{x}_0$	$\bar{x}_0$	
y_6	0 1 1 0	Antivalenz (XOR,Exclusiv-Oder)	$x_1\bar{x}_0 \vee \bar{x}_1 x_0$	$(\bar{x}_1 \vee \bar{x}_0)(x_1 \vee x_0)$	$x_1 \not\sim x_0$	
y_7	0 1 1 1	NAND (not and)	$\bar{x}_1 \vee \bar{x}_0$	$\bar{x}_1 \vee \bar{x}_0$	$\overline{x_1 x_0}$	
y_8	1 0 0 0	AND (Konjunktion)	$x_1 x_0$	$x_1 x_0$	$x_1 x_0$	
y_9	1 0 0 1	Äquivalenz	$x_1 x_0 \vee \bar{x}_1\bar{x}_0$	$(\bar{x}_1 \vee x_0)(x_1 \vee \bar{x}_0)$	$x_1 \sim x_0$	
y_{10}	1 0 1 0	Identität von x_0	x_0	x_0	x_0	
y_{11}	1 0 1 1	Implikation von x_1 auf x_0	$\bar{x}_1 \vee x_0$	$\bar{x}_1 \vee x_0$	$x_1 \rightarrow x_0$	
y_{12}	1 1 0 0	Identität von x_1	x_1	x_1	x_1	
y_{13}	1 1 0 1	Implikation von x_0 auf x_1	$x_1 \vee \bar{x}_0$	$x_1 \vee \bar{x}_0$	$x_0 \rightarrow x_1$	
y_{14}	1 1 1 0	OR (Disjunktion)	$x_1 \vee x_0$	$x_1 \vee x_0$	$x_1 \vee x_0$	
y_{15}	1 1 1 1	Eins	1	1	1	

Beispiele für kombinatorische Strukturen

(1) Schaltsymbole auf Seite 09

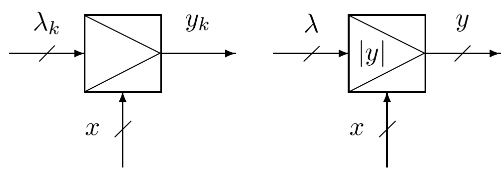
(2) Multiplexer

interne Struktur



interne Struktur des Multiplexers k

Symbol

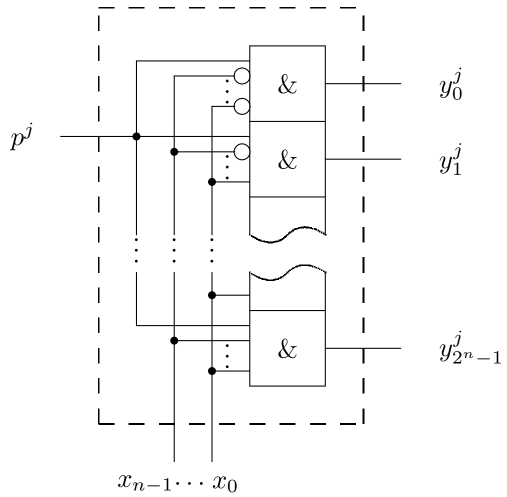


Multiplexer k

Multiplexerblock
aus $|y|$ Multiplexern

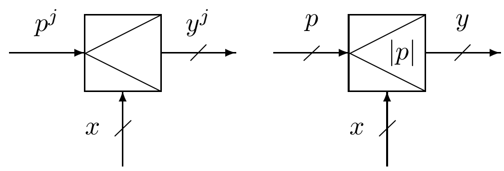
(3) Demultiplexer

interne Struktur



interne Struktur des Demultiplexers j

Symbol



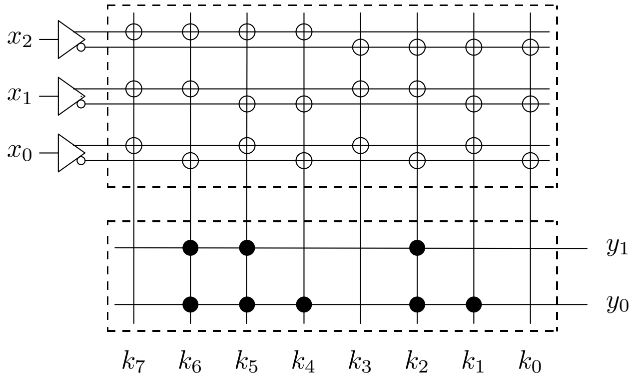
Demultiplexer j

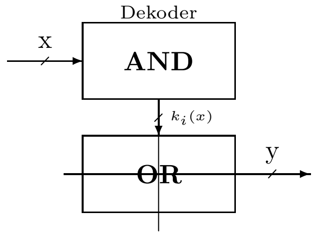
Demultiplexerblock
aus $|p|$ Demultiplexern

(4) Programmierbare Strukturen (Hinweis: AND- und OR-Matritzen können auch als NOR-Matritzen realisiert sein)

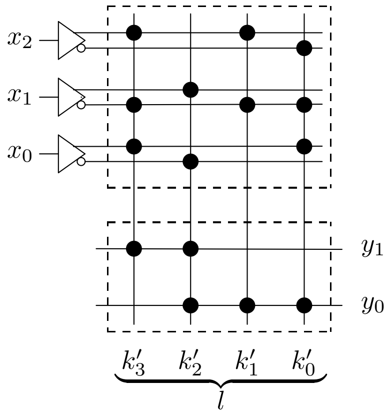
Beispiel: $y_1 = x_2\bar{x}_1x_0 \vee x_1\bar{x}_0 \quad y_0 = x_2\bar{x}_1 \vee \bar{x}_2\bar{x}_1x_0 \vee x_1\bar{x}_0$

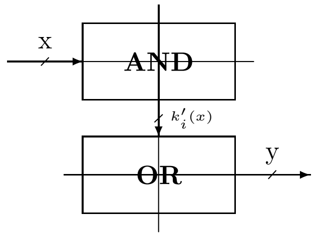
ROM Ausgangspunkt: Elementarkonjunktionen $k_i(x)$ bzw. KDNF



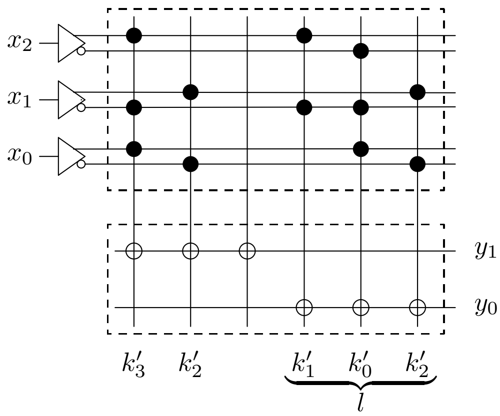


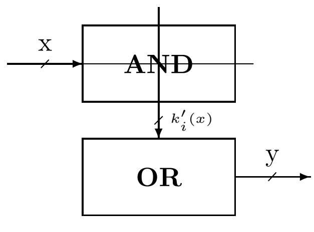
PLA Ausgangspunkt: Gleichungen in DNF mit max. l unterschiedlichen Fundamentalkonjunktionen für alle Gleichungen

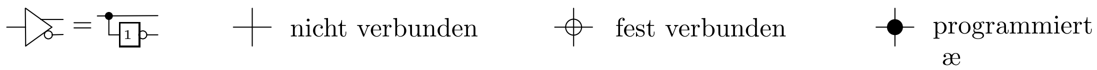




PAL/GAL Ausgangspunkt: Gleichungen in DNF mit max. l Fundamentalkonjunktionen je Gleichung







TU Ilmenau
Institut TII
FG IHS

Zusammenfassung Beschreibungsmittel

Synthese

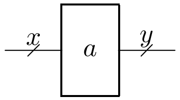
Funktion $\rightleftarrows$ Schaltung

Analyse

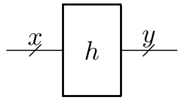
BMA	**BAA**	**BAA**	**Struktur**
$\lambda:\ X(x) \Rightarrow Y(y)$	$y = a(x)$	$y\ =\ h(x)$	$S = [M, x, p, y, \text{æ}]$

x	y
X	Y

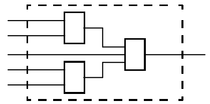
$\xLeftrightarrow{\text{Wertfunktion}}$



$\xLeftrightarrow{\text{Auswahl}}$



$\xLeftrightarrow{\text{Modulzuordnung}}$



Wertetabelle	**Gleichungssystem**	**strukturorientiertes Gleichungssystem**	**Schaltungsstruktur aus gekoppelten Moduln**
mit $\lvert X\rvert + \lvert Y\rvert$ Spalten und $2^{\lvert x\rvert} = \lvert X\rvert$ Zeilen	aus $\lvert y\rvert$ Gleichungen	aus $\lvert y\rvert$ Gleichungen	

Wertetabelle:

x_2	x_1	x_0	y_1	y_0
0	0	0	1	1
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Gleichungssystem:

$y_0 = k_0 \vee k_1 \vee k_4$

$y_1 = k_0 \vee k_3 \vee k_4 \vee k_5 \vee k_6$

y_0	$\bar{x}_1\bar{x}_0$	$\bar{x}_1 x_0$	$x_1 x_0$	$x_1\bar{x}_0$
$\bar{x}_2$	1 (k_0)	1 (k_1)	0 (k_3)	0 (k_2)
x_2	1 (k_4)	0 (k_5)	0 (k_7)	0 (k_6)

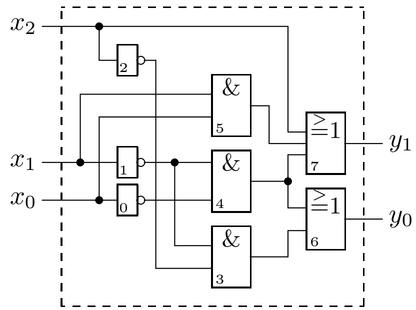
y_1	$\bar{x}_1\bar{x}_0$	$\bar{x}_1 x_0$	$x_1 x_0$	$x_1\bar{x}_0$
$\bar{x}_2$	1 (k_0)	0 (k_1)	1 (k_3)	0 (k_2)
x_2	1 (k_4)	1 (k_5)	1 (k_7)	1 (k_6)

Strukturorientiertes Gleichungssystem:

$$y_{0_{min}} = \underbrace{\underbrace{\underbrace{\bar{x}_1}_{M^1}\ \bar{x}_0}_{M^4} \vee \underbrace{\bar{x}_2\bar{x}_1}_{M^3}}_{M^6}$$

$$y_{1_{min}} = \underbrace{x_2 \vee \underbrace{\bar{x}_1\bar{x}_0}_{M^4} \vee \underbrace{x_1 x_0}_{M^5}}_{M^7}$$

Schaltungsstruktur:



$y_0 \underset{\circ}{=} y_{0_{min}} \qquad y_1 \underset{\circ}{=} y_{1_{min}}$

$M = \{M_0, M_1, \ldots, M_7\}$

$x = [x_2, x_1, x_0]$

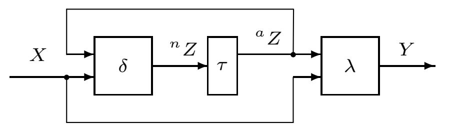
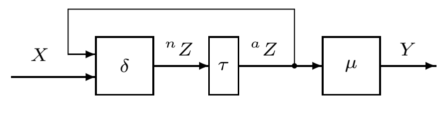
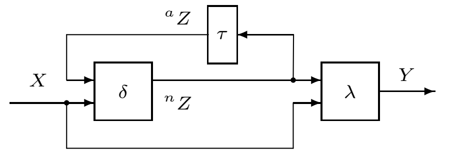
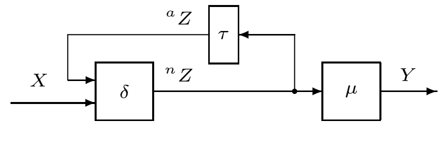
$p = \emptyset$

$y = [y_1, y_0]$

$\text{æ}(y_1) = y_0^7, \ldots, \text{æ}(x_0^0) = x_0$

Sequentielle Automaten

Automatentypen

	Mealy-Automat $A_\lambda = [X, Y, Z, \delta, \lambda]$	**Moore-Automat** $A_\mu = [X, Y, Z, \delta, \mu]$
Typ a	 $\lambda : {}^aZ \times X \Rightarrow Y$	 $\mu : {}^aZ \Rightarrow Y$
Typ n	 $\lambda : {}^nZ \times X \Rightarrow Y$	 $\mu : {}^nZ \Rightarrow Y$

$X \ldots$ Eingangsbelegungsmenge des Eingangsvektors $x = [x_{n-1}, \ldots, x_r, \ldots, x_0]$

$Y \ldots$ Ausgangsbelegungsmenge des Ausgangsvektors $y = [y_{m-1}, \ldots, y_k, \ldots, y_0]$

$Z \ldots$ Zustandsbelegungsmenge z.B. $Z_1(z) = [0, 0, \ldots, 0, 1] \in Z$ mit $z = [z_{p-1}, \ldots, z_s, \ldots, z_0]$ als p-stelliger Zustandsvektor z aus p binären Zustandsvariablen

- ${}^nZ \ldots$ neuer Zustand
- ${}^aZ \ldots$ alter Zustand

$\tau \ldots$ Verzögerung

$\delta \ldots$ Zustandsüberführungsfunktion $\delta : Z \times X \Rightarrow Z$

$\lambda / \mu \ldots$ Mealy-/Moore-Ausgabefunktion

Automatentabelle für einen Mealy-Automaten vom Typ a

Belegungsindex						0	$\cdots$	i	$\cdots$	$2^n - 1$
					x_0	0	$\cdots$	$X_i(x_0)$	$\cdots$	1
					$\vdots$					
					x_r	0	$\cdots$	$X_i(x_r)$	$\cdots$	1
					$\vdots$					
					x_{n-1}	0	$\cdots$	$X_i(x_{n-1})$	$\cdots$	1
	z_{p-1}	$\cdots$	z_s	$\cdots$	z_0					
0	0	$\cdots$	0	$\cdots$	0					
$\vdots$										
j	${}^aZ_j(z_{p-1})$	$\cdots$	${}^aZ_j(z_s)$	$\cdots$	${}^aZ_j(z_0)$			${}^nZ_u := \delta({}^aZ_j, X_i)$ $Y_t = \lambda({}^aZ_j, X_i)$		
$\vdots$										
$2^p - 1$	1	$\cdots$	1	$\cdots$	1					

Zustands- und Automatengraphen

Zustandsgraph	$G_\delta = [Z, K, \omega_\delta]$
Mealy-Automatengraph	$G_\lambda = [Z, K, \omega_\delta, \omega_\lambda]$
Moore-Automatengraph	$G_\mu = [Z, K, \omega_\delta, \omega_\mu]$

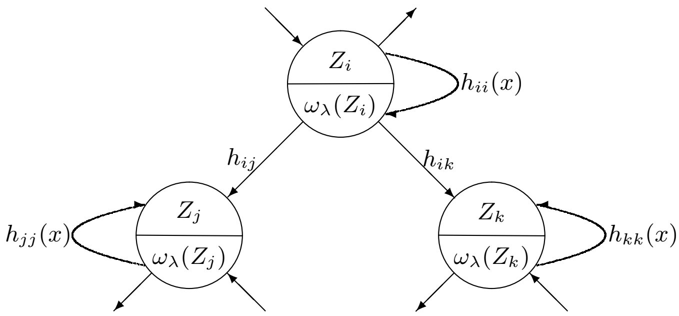
mit:

- $Z \ldots$ Knotenmenge
- $K \ldots$ Kantenmenge mit $K \subseteq Z \times Z$
- $\omega_\delta \ldots$ Kantengewichtsfunktion
- ω_λ bzw. $\omega_\mu \ldots$ Knotengewichtsfunktionen

wobei im einzelnen für Zustände und Kanten gilt:

- $[Z_i, Z_j] \in K \leftrightarrow \overline{h_{ij}(x) = 0}$
- $\omega_\delta([Z_i, Z_j]) = h_{ij}(x)$
  $h_{ij}(x) \ldots$ Übergangsausdruck (Kantengewicht der Kante $[Z_i, Z_j]$)
- $\omega_\lambda(Z_i) = \{y_k = h_{ik}(x) \quad | \quad 0 \le k \le m-1\}$ (Mealy)
- $\omega_\mu(Z_i) = \{y_k = h_{ik}(0,1) \quad | \quad 0 \le k \le m-1\}$ (Moore)
  $h_{ik}(x) \ldots$ Ausgabeausdruck der y_k-Komponente in Z_i, $m = |y|$

Allgemeine graphische Notationsform für G_λ



Vollständigkeit und Widerspruchsfreiheit

Vollständigkeit	$\forall i (\bigvee_{j=0}^{2^p-1} h_{ij}(x) = 1)$
Widerspruchsfreiheit	$\forall i (\bigvee_{\substack{j,k=0 \\ j \neq k}}^{2^p-1} h_{ij}(x) \wedge h_{ik}(x) = 0)$

Für die schaltalgebraische Realisierung von Automaten kann jeder Zustand Z_i als Belegung Z_i des Zustandsvektors $z = [z_{p-1}, \ldots, z_0]$ interpretiert und durch eine Elementarkonjunktion von Zustandsvariablen $k_i(z)$ repräsentiert werden.

Aus der Automatentabelle lassen sich ableiten:

1. die Zustandsüberführungsfunktion δ

- **als Gleichungen für die Elementarkonjunktionen k_j der Zustände Z_j**

$$k_j(z) := \bigvee_{i=0}^{2^p-1} k_i(z) \wedge h_{ij}(x)$$

Anmerkung: h_{ij} ist ein Ausdruck in x-Variablen, der die Menge aller Eingangsbelegungen X_k repräsentiert, für die ein Zustandsübergang von Z_i nach Z_j erfolgt.

$\mathcal{W}(h_{ij}, X_k) = 1 \Leftrightarrow \delta(Z_i, X_k) = Z_j$

- **oder als Gleichungen für die Zustandsvariablen $z_k := h_\delta(z, x)$**

$$z_k := \bigvee_{j \in M_k} \left(\bigvee_{i=0}^{2^p-1} k_i(z) \wedge h_{ij}(x) \right) \qquad \text{mit} \quad j \in M_k \leftrightarrow Z_j(z_k) = 1$$

2. die Ausgabefunktion λ (bzw. μ)

- **für Mealy-Automaten $y_k = h_\lambda(z, x)$**

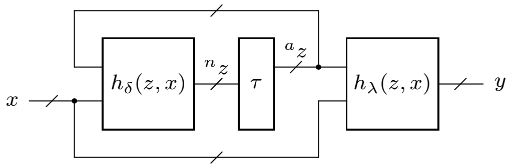
$$y_k = \bigvee_{i=0}^{2^p-1} k_i(z) \wedge h_{ik}(x)$$

- **für Moore-Automaten $y_k = h_\mu(z)$**

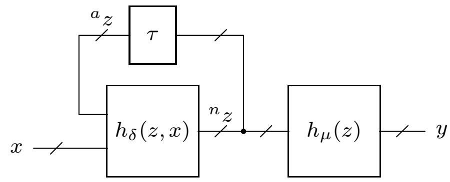
$$y_k = \bigvee_{i=0}^{2^p-1} k_i(z) \wedge h_{ik}(0, 1)$$

Struktursynthese sequentieller Automaten

Die direkte Strukturinterpretation der Zustands- und Ausgabegleichungen (siehe Seite 32) liefert sogenannte **asynchrone** sequentielle Schaltungen mit folgender Grobstruktur:



Mealy



Moore

Eine sequentielle Struktur heißt **synchron**, wenn alle Belegungswechsel ihrer Eingangs-/Ausgangs- und Zustandsvariablen nur zu definierten – über einen zentralen Takt steuerbaren – Zeitpunkten (oder -intervallen) funktionsrelevant sind und für mindestens eine halbe Taktperiode gespeichert bleiben.

Die Taktfrequenz in synchronen Strukturen ist so zu wählen, daß alle Hasards sicher beendet sind.

Die Synchronisation und Zwischenspeicherung "alter" Belegungen erfolgt bitweise in bistabilen Kippschaltungen, sogenannten Flop-Flips, die die symbolische Verzögerung τ in den oben gezeigten Strukturbildern ersetzen.

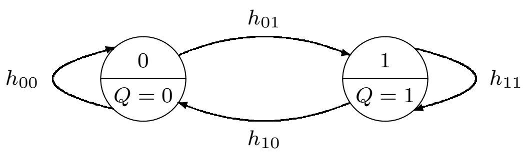
Struktursynthese mit Flip-Flops:

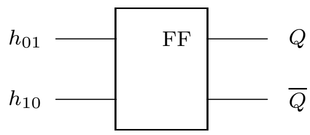
1. binäre Zustandskodierung
   - $|z| = n = \lceil ld|Z| \rceil \quad \Rightarrow \quad z = [z_{n-1}, \ldots, z_i, \ldots, z_0]$ als n-stelliger Zustandsvariablenvektor
2. Ermittlung der Ansteuergleichungen für Flip-Flops durch
   (a) Aufstellen der z-Gleichungen (siehe Seite 32)
   (b) Umformen der z-Gleichungen in Form der charakteristischen Gleichung der Ziel-Flip-Flops
   $z_i := z_i\overline{h_{10}}(x, z\backslash z_i) \vee \overline{z_i}h_{01}(x, z\backslash z_i)$
   (c) Ermitteln der Ansteuergleichungen durch Gleichsetzen der Koeffizienten $\overline{h_{10}}$ bzw. h_{01} der z_i-Variablen mit den Koeffizienten der Q-Variablen der charakteristischen Gleichung (z.B.: $J = h_{01},\ K = \overline{h_{10}}$ – siehe Seite 41)

   oder direktes Auslesen aus dem Graph:
   - Disjunktionen der Kantengewichte der $0-1$-Übergänge von z_i für h_{01} und der $1-0$-Übergänge für h_{10}
3. Ermittlung der Ausgabegleichungen
   - siehe Seite 32

Flip-Flops

Flip-Flops sind elementare sequentielle Strukturen, deren Funktion abstrahiert mit zwei stabilen Zuständen beschreibbar ist:





Falls δ vollständig und widerspruchsfrei ist, gilt:

- $h_{00} = \overline{h_{01}}$ und $h_{11} = \overline{h_{10}}$

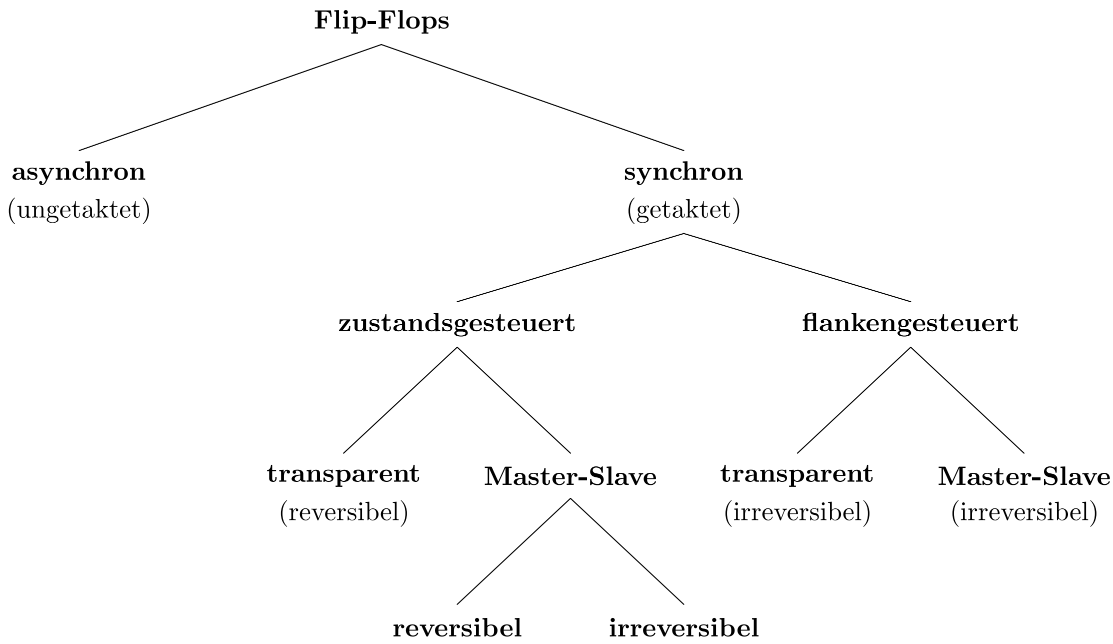
und somit:

- $\delta: \quad z := z\,\overline{h_{10}} \;\vee\; \overline{z}\,h_{01}$
- $\mu: \quad Q = z$

Das Ein-/Ausgangsverhalten eines Flip-Flops läßt sich mit folgender Gleichung charakterisieren:

- $Q := Q\,\overline{h_{10}} \;\vee\; \overline{Q}\,h_{01}$ (charakteristische Gleichung)

Klassifikation der Flip-Flops:

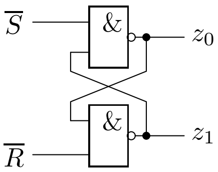


(I) Basis-FF (asynchron)

Für das RS-Flip-Flop mit

- $h_{01} = S$ und $h_{10} = R$

ist folgende vereinfachte Schaltung praxisrelevant, bei der die beiden Gatterausgänge Q und $\overline{Q}$ als Zustandsvariablen z_0 und z_1 betrachtet werden:

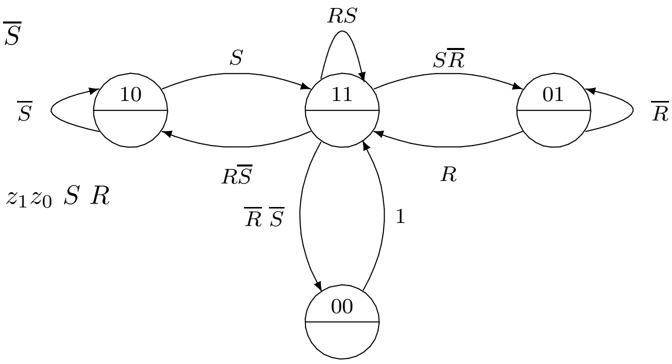


Daraus ergeben sich die z-Gleichungen:

- $z_0 := \overline{\overline{S} \wedge z_1}$
- $z_1 := \overline{\overline{R} \wedge z_0}$

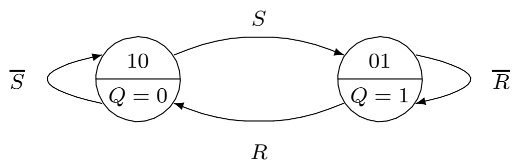
und daraus die Zustandsgleichungen für den **resultierenden Automatengraphen**:

- $\overline{z_1}\ \overline{z_0} := \overline{\overline{\overline{R} \wedge z_0}} \wedge \overline{\overline{\overline{S} \wedge z_1}} = z_1 z_0\ \overline{R}\ \overline{S}$
- $\overline{z_1}\ z_0 := \overline{z_1} z_0\ \overline{R} \vee z_1 z_0\ S\ \overline{R}$
- $z_1\ \overline{z_0} := z_1 \overline{z_0}\ \overline{S} \vee z_1 z_0\ \overline{S}\ R$
- $z_1\ z_0 := \overline{z_1}\ \overline{z_0} \vee \overline{z_1} z_0\ R \vee z_1 \overline{z_0}\ S \vee z_1 z_0\ S\ R$



Über folgende Abstraktion läßt sich der **abstrahierte Automatengraph** ableiten:
Da $Q \neq \overline{Q}$ gelten soll, müssen Z_0 und Z_3 und damit die Eigenschleife von Z_3 verboten werden.
Daraus folgt $h^* = R\ S$ und:

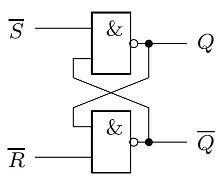
- für $S = 1$ ein Wechsel von Z_2 nach Z_1
- für $R = 1$ ein Wechsel von Z_1 nach Z_2



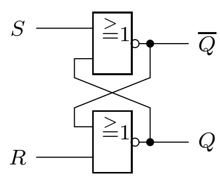
char. Gleichung: $Q := Q\ \overline{R}\ \vee\ \overline{Q}\ S$

Schaltung:

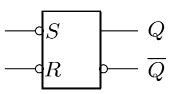
RS-NAND-FF

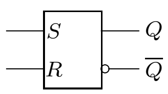


RS-NOR-FF



Symbol:

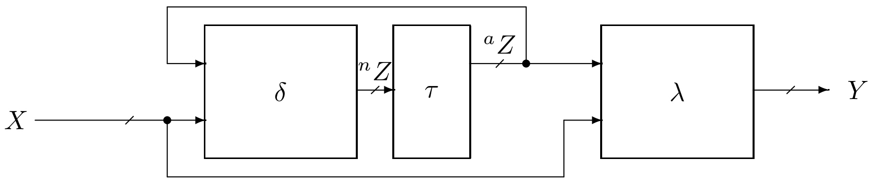




Entwurf paralleler Automaten

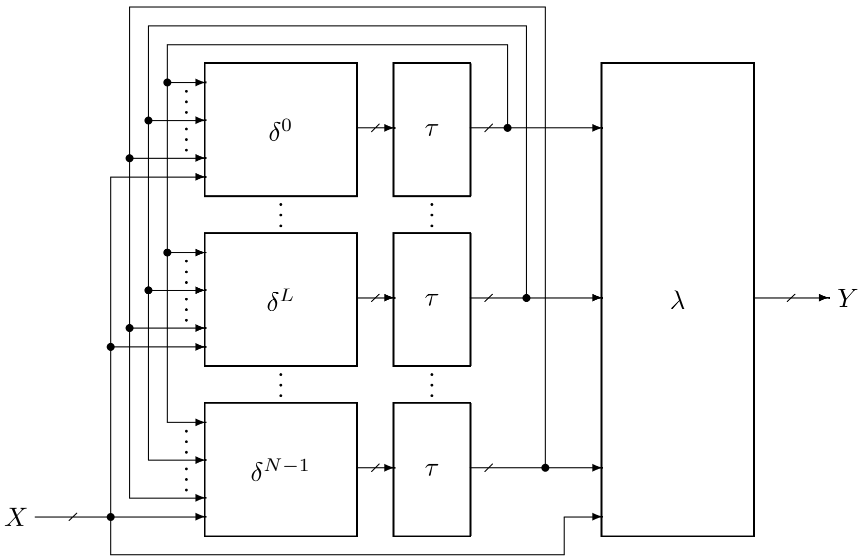
Funktionelle Dekomposition

Gegeben: Automat $A = (X, Z, Y, \delta, \lambda, Z^{\bullet})$ $\qquad Z^{\bullet} \in Z$, Initialzustand



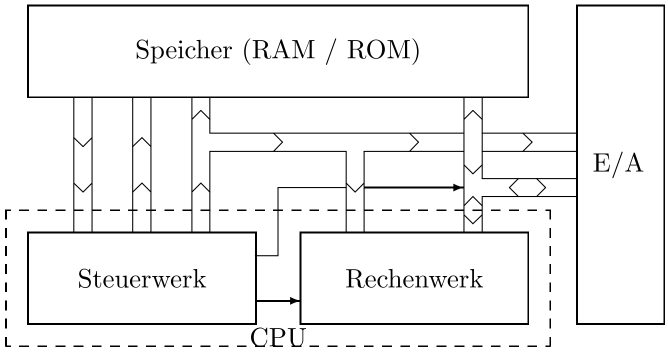
Automat A

Gesucht: Dekomposition in N Teilautomaten $A^0, A^1, \ldots, A^L, \ldots, A^{N-1}$
nach semantischen Kriterien entsprechend der angestrebten Teilfunktionen

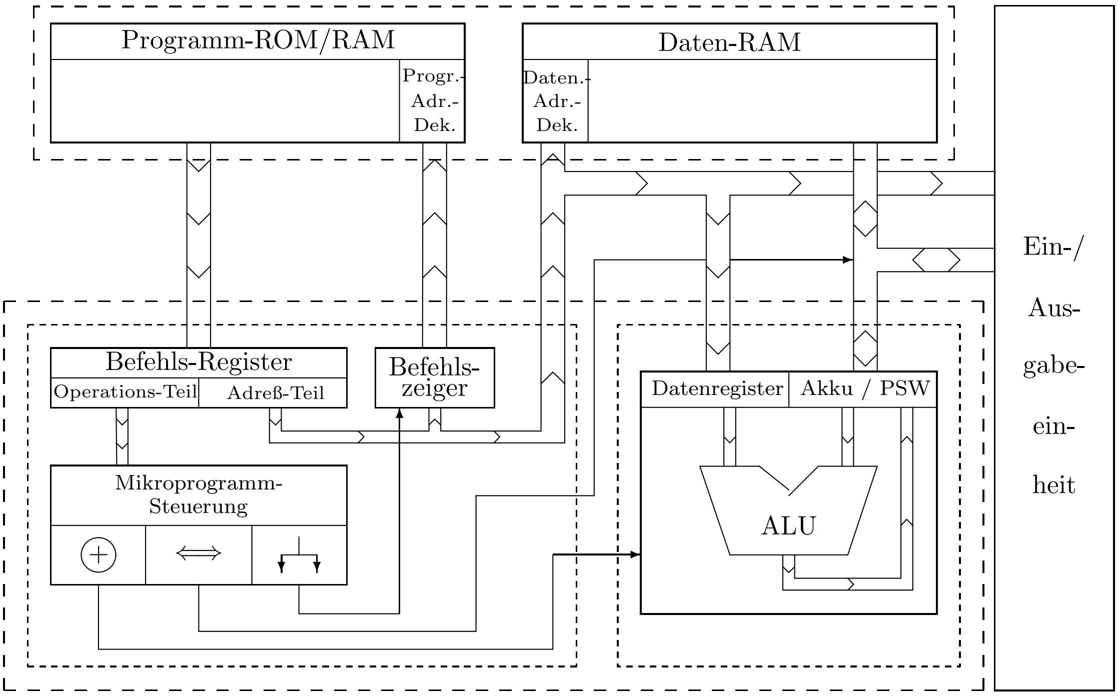


Mikrorechner-Architektur

Allgemeine Struktur



Detaillierte Struktur



Harvard-Architektur	:	Programm- und Datenspeicher getrennt
von-Neumann-Architektur	:	Programm- und Datenspeicher gemeinsam

Informationskodierung $\mathcal{K} : \boldsymbol{Vb} \Longleftrightarrow \boldsymbol{Nb}$

Vorbereich (Vb)	Befehle, Daten
Nachbereich (Nb)	Menge von n-Tupeln binärer Werte, wobei $N_n = \{0,1\}^n$ ... Menge aller n-Tupel
n-stelliger Kode	k heißt n-stelliger Kode von z, falls $z \in$ Vb, $k \in N_n$ und $\mathcal{K}(z) = k$
Kodiervorschrift KV	Eine Kodiervorschrift besteht aus Angaben zur Struktur sowie einem Algorithmus für die Kode-Erzeugung.

I. Zeichen-Kodierung

ASCII-Kodierung

- Vb: α-numerische Zeichen (in der Tabelle: Spalte α-n)
- Nb: N_8 ... Menge von Bytes (dargestellt als Hexadezimalzahlen $(ASC)_{16}$ bzw. Dezimalzahlen $(ASC)_{10}$)
- Kodiervorschrift: siehe nachfolgende Tabelle

$(ASC)_{16}$	$(ASC)_{10}$	α-n	$(ASC)_{16}$	$(ASC)_{10}$	α-n	$(ASC)_{16}$	$(ASC)_{10}$	α-n	$(ASC)_{16}$	$(ASC)_{10}$	α-n
00	00	NUL	20	32	SP	40	64	@	60	96	‘
01	01	SOH	21	33	!	41	65	A	61	97	a
02	02	STX	22	34	”	42	66	B	62	98	b
03	03	ETX	23	35	#	43	67	C	63	99	c
04	04	EOT	24	36	$	44	68	D	64	100	d
05	05	ENQ	25	37	%	45	69	E	65	101	e
06	06	ACK	26	38	&	46	70	F	66	102	f
07	07	BEL	27	39	’	47	71	G	67	103	g
08	08	BS	28	40	(	48	72	H	68	104	h
09	09	HT	29	41	)	49	73	I	69	105	i
0A	10	LF	2A	42	∗	4A	74	J	6A	106	j
0B	11	VT	2B	43	+	4B	75	K	6B	107	k
0C	12	FF	2C	44	,	4C	76	L	6C	108	l
0D	13	CR	2D	45	-	4D	77	M	6D	109	m
0E	14	SO	2E	46	.	4E	78	N	6E	110	n
0F	15	SI	2F	47	/	4F	79	O	6F	111	o
10	16	DLE	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1A	26	SUB	3A	58	:	5A	90	Z	7A	122	z
1B	27	ESC	3B	59	;	5B	91	[	7B	123	{
1C	28	FS	3C	60	<	5C	92	\	7C	124	|
1D	29	GS	3D	61	=	5D	93	]	7D	125	}
1E	30	RS	3E	62	>	5E	94	ˆ	7E	126	˜
1F	31	US	3F	63	?	5F	95	_	7F	127	DEL

II. Zahlen-Kodierung

(1) BCD-Zahlen

- Vb: Z ... Menge der Dezimalziffern z_i mit $z_i \in \{0, 1, 2, \ldots, 9\}$
- Nb: $T = N_4 \backslash P$ mit:
  - $T \ldots$ Menge der Tetraden t_j
  - $P \ldots$ Menge der Pseudotetraden p
- KV: $\mathcal{K}(z_i) = t_j$ (Indexbestimmung: siehe Tabelle)

Ausgewählte BCD-Kodes

z_i (i ... Dezimalziffernwert)				t_j	
BCD (direkt)	**Aiken**	**3xS**	**Gray**	$2^3 2^2 2^1 2^0$	j
0	0	-	0	0 0 0 0	0
1	1	-	1	0 0 0 1	1
2	2	-	3	0 0 1 0	2
3	3	0	2	0 0 1 1	3
4	4	1	7	0 1 0 0	4
5	-	2	6	0 1 0 1	5
6	-	3	4	0 1 1 0	6
7	-	4	5	0 1 1 1	7
8	-	5	9	1 0 0 0	8
9	-	6	-	1 0 0 1	9
-	-	7	-	1 0 1 0	A
-	5	8	-	1 0 1 1	B
-	6	9	8	1 1 0 0	C
-	7	-	-	1 1 0 1	D
-	8	-	-	1 1 1 0	E
-	9	-	-	1 1 1 1	F
$j := i$	(a)	$j := i + 3$		$j \ldots Dualzahlenwert$	

(a): $j := \begin{cases} i & \text{falls} \quad 0 \leq i \leq 4 \\ i+6 & \text{falls} \quad 5 \leq i \leq 9 \end{cases}$

Operationen mit BCD-Zahlen

BCD-Zahlen werden bei Addition und Subtraktion entsprechend der Gesetze der Dualzahlen-Arithmetik verknüpft.
Da hierbei je Dezimalziffer der Zahlenbereich vierstelliger Dualzahlen genutzt wird, sind die Ergebnisse in Abhängigkeit von auftretenden **Pseudotetraden p** bzw. **Tetradenüberträgen ü** kodeabhängig mit Hilfe einer **Konstanten C** folgendermaßen zu korrigieren:

- bei Addition $\mathcal{K}(z_i + z_i') = \mathcal{K}(z_i) + \mathcal{K}(z_i') + \mathbf{C}$
- bei Subtraktion $\mathcal{K}(z_i - z_i') = \mathcal{K}(z_i) - \mathcal{K}(z_i') - \mathbf{C}$

	BCD	Aiken	3xS
Korrektur erforderlich	bei **ü** oder **p**	nur bei **p**, dann abhängig von **ü**	immer, aber abhängig von **ü**
Konstante C	0110	−0110 bei **ü** +0110 sonst	+0011 bei **ü** −0011 sonst

(2) Vorzeichen-Betrags-Zahlen

- Vb: $-2^{n-1} < z < 2^{n-1}$
- Nb: N_n
- KV: $MSB = 0 \quad \Leftrightarrow \quad z \geq 0$
  $MSB = 1 \quad \Leftrightarrow \quad z \leq 0$
  $Rest := |z_{n-1}|$

- Struktur:

MSB	$Betrag$		
2^{n-1}	2^{n-2}	...	2^0

MSB ... Most Significant Bit
$Betrag$ ... $n-1$-stellige Darstellung der Dualzahl von $|z|$

(3) Konegative Zahlen

- Vb: $-2^{n-1} < z < 2^{n-1}$
- Nb: N_n (üblich: N_8, N_{16}, N_{32})
- KV:
  $\mathcal{K} = (z) = \begin{cases} z_n \leftrightarrow z \geq 0 \\ \overline{z_n} \text{ sonst} \end{cases}$

  z_n ... n-stellige Dualzahl von z
  $\overline{z_n}$ ... Komplement der Dualzahl

Komplementbildung

Man unterscheidet Einer- und Zweierkomplemente und dementsprechend als konegative Zahlen 1K- und 2K-Zahlen

- 1K-Zahlen: $\overline{z_n} = 2^n - 1 - z_n$
- 2K-Zahlen: $\overline{z_n} = 2^n - z_n$

Operationen mit Konegativen Zahlen

Für die Addition bzw. Subtraktion zweier Dualzahlen z_{n1} und z_{n2} mit

- $z_{n1} \geq z_{n2}$
- $s_n = z_{n1} + z_{n2}$ als Summe
- $d_n = z_{n1} - z_{n2}$ als Differenz

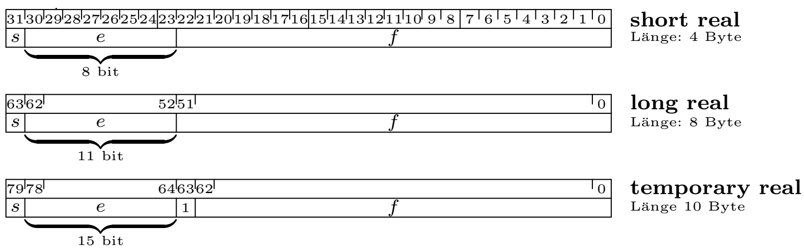
gilt bei Ersetzung der Komplemente entsprechend den Vorschriften zur Komplementbildung:

Operation	1k-Zahlen		2K-Zahlen	
	Ergebnis	Korrektur	Ergebnis	Korrektur
$z_{n1} + z_{n2}$	s_n	-	$z_{n1} + z_{n2} = s_n$	-
$z_{n1} + \overline{z_{n2}}$	$d_n + 2^n - 1$	$-2^n + 1$	$z_{n1} - z_{n2} + 2^n = d_n + 2^n$	-2^n
$\overline{z_{n1}} + z_{n2}$	$\overline{d_n}$	-	$-(z_{n1} - z_{n2}) + 2^n = \overline{d_n}$	-
$\overline{z_{n1}} + \overline{z_{n2}}$	$\overline{s_n} + 2^n - 1$	$-2^n + 1$	$-(z_{n1} + z_{n2}) + 2^n + 2^n = \overline{s_n} + 2^n$	-2^n

(4) Gleitkomma-Zahlen (IEEE-Standard)

- Vb: Menge rationaler Zahlen z
- Nb: N_{32} ...short, N_{64} ...long, N_{80} ...temporary
- KV: s ... sign (Vorzeichen der Zahl)
  e ... biased exponent (vorzeichenloser Exponent)
  f ... fractional part (gebrochener Anteil)

- Struktur:



Kodierung

1. Bestimmung von s $\quad s = \begin{cases} 1 & \Leftrightarrow z < 0 \\ 0 & \text{sonst} \end{cases}$
2. Umwandlung von z in eine Dualzahl $\quad |z| \rightarrow z_n$
3. normierte halblogarithmische Darstellung von $z_n = M \cdot (10)_2{}^E$
   mit: M ... Mantisse, E ... Exponent, so daß gilt: $1 \leq M < (10)_2$
4. Abspaltung von f aus $M = 1, f$
5. Berechnung von $e = E + (bias)_2 \quad (bias)_{16} = \begin{cases} 7F & \text{für short} \\ 3FF & \text{für long} \\ 3FFF & \text{für temporary} \end{cases}$
6. Formatanpassung gemäß der Struktur

Wertebereich

Datentyp	Bits	Wertebereich	Genauigkeit
short real	32	$\pm 1,2 * 10^{-38} \ldots \pm 3,4 * 10^{38}$	
long real	64	$\pm 2,2 * 10^{-308} \ldots \pm 1,8 * 10^{308}$	
temporary real	80	$\pm 1,1 * 10^{-4932} \ldots \pm 1,2 * 10^{4932}$	

spezielle Werte (für N_{32})

s	e	f	Wert
1/0	$= 00$	$= 0$	± 0
1/0	$= 00$	$\neq 0$	denormalisiert
1/0	$= FF$	$= 0$	$\pm\infty$
1/0	$= FF$	$\neq 0$	NaN (not a number)

Operationen

- getrennte Vorzeichen-, Exponenten- und Mantissenberechnung
- Exponentenanpassung bei Addition und Subtraktion
- Rundungen

Anhang

Mathematische Grundlagen

Aussagen

Definitionen

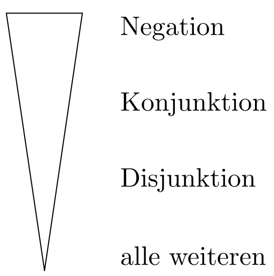
elementare Aussagen	sind Sätze zur Beschreibung von Eigenschaften (Prädikaten) p, q, ... von Individuen $x_0, x_1, \ldots$ aus einem bestimmten Individuenbereich $x = \{x_0, x_1, \ldots\}$. z.B.: x_0 hat die Eigenschaft p. Aussagen sind entweder $wahr(w)$ oder $falsch(f)$.
Aussagenvariable	$A, B, \ldots$ sind Symbole für Aussagen.
aussagenlogische Ausdrücke	sind folgendermaßen induktiv definiert: 1. die Aussagenvariablen $A, B, \ldots$ und die Wahrheitswerte w und f sind Ausdrücke, 2. wenn A und B Ausdrücke sind, so sind auch $\overline{A}, (A \wedge B), (A \vee B), (A \rightarrow B)$ und $(A \leftrightarrow B)$ Ausdrücke, 3. nur die nach (1) und (2) gebildeten Zeichenketten sind Ausdrücke.

Elementare und zusammengesetzte Aussagen lassen sich mit Hilfe aussagenlogischer Ausdrücke beschreiben.

Wahrheitswerte zusammengesetzter Aussagen

A	B	Negation (A nicht) $\overline{A}$	Negation (B nicht) $\overline{B}$	Konjunktion (A und B) $A \wedge B$	Disjunktion (A oder B) $A \vee B$	Implikation (wenn A dann B) $A \rightarrow B$	Äquivalenz (A genau dann wenn B) $A \leftrightarrow B$
f	f	w	w	f	f	w	w
f	w	w	f	f	w	w	f
w	f	f	w	f	w	f	f
w	w	f	f	w	w	w	w

Priorität



Wichtige *Äquivalenzen* aussagenlogischer Ausdrücke:

$$
\begin{array}{lll}
A \rightarrow B & \text{äquivalent} & \overline{A} \vee B \\
A \leftrightarrow B & \text{äquivalent} & (A \wedge B) \vee (\overline{A} \wedge \overline{B}) \\
\overline{A \wedge B} & \text{äquivalent} & \overline{A} \vee \overline{B} \\
\overline{A \vee B} & \text{äquivalent} & \overline{A} \wedge \overline{B}
\end{array}
$$

Kontradiktion ist ein aussagenlogischer Ausdruck, welcher nach beliebiger Wertzuweisung zu Aussagevariablen immer den Wert f hat.
Beispiele:

- $A \wedge \overline{A}$
- $A \wedge f$
- $(A \rightarrow B) \wedge (A \wedge \overline{B})$

Tautologie ist ein aussagenlogischer Ausdruck, welcher nach beliebiger Wertzuweisung zu Aussagenvariablen immer den Wert w hat.
Beispiele:

- $A \vee \overline{A}$
- $A \vee w$
- $(A \rightarrow B) \vee A$

HORN – Klauseln sind aussagenlogische Ausdrücke der Form:

- $A_1 \wedge A_2 \wedge \ldots A_n \rightarrow B$ "Regel"
- $w \rightarrow B$ "Fakt"

Notizen

Prädikate

Prädikat — Teil einer Aussage, der eine klassifizierende Eigenschaft $(p, q, \ldots)$ beinhaltet.

abhängige Aussage — Der Wahrheitswert abhängiger Aussagen (z.B. $p(x), q(y), \ldots$) kann erst bestimmt werden, wenn die Individuensymbole (z.B. $x, y, \ldots$) durch konkrete Individuen eines Individuenbereiches ersetzt werden.

	abhängige Aussage	*Individuenbereich von x,y*
$p(x)$	x ist durch 25 teilbar	ganze Zahlen
$q(y)$	y ist ein Sommertag	alle Tage des Dezember
$r(x, y)$	x ist Hauptstadt von y	alle Städte > 1Mio. EW

wobei z.B. p das Symbol für das Prädikat *"ist durch 25 teilbar"* darstellt

Prädikatenlogische Ausdrücke und deren Sprechweise

Durch Einbeziehung der Individuensymbole in den Wirkungsbereich von Quantifikatoren (Allquantor $\forall$, *Existenzquantor* $\exists$) können prädikatenlogische Ausdrücke formuliert werden.

(a) $\exists x(r(x))$ — es existiert (mindestens) ein Element im Individuenbereich von x, für das gilt, das Prädikat r ist für x *wahr* (sprich: *"r von x ist wahr"*).

(b) $\overline{\forall x(\overline{r(x)})}$ — es gilt nicht, daß für alle Elemente aus dem Individuenbereich von x $r(x)$ nicht *wahr* ist – gleicher Sachverhalt wie unter (a);

(c) $\forall y(s(y)) = \overline{\exists y(\overline{s(y)})}$ — für alle Elemente des Individuenbereiches von y gilt, die Aussage $s(y)$ ist *wahr* bzw.: es gibt kein y, für das $s(y)$ nicht wahr ist;

(d) $\forall y \exists x(t(x, y))$ — für alle Elemente des Individuenbereiches von y existiert (mindestens) ein Element aus dem Individuenbereich von x, für welches gilt, daß $t(x, y)$ *wahr* ist;

(e) $\exists y \forall x(r(x) \rightarrow s(y))$ — es existiert (mindestens) ein Element im Individuenbereich von y, für das gilt, daß für alle Elemente des Individuenbereiches von x die Implikation $r(x) \rightarrow s(y)$ (sprich: *"aus $r(x)$ folgt $s(y)$"*) *wahr* ist.

In allen gezeigten Beispielen für prädikatenlogische Ausdrücke kommen die Individuensymbole x und y *gebunden* vor, da sie im Wirkungsbereich des *Allquantors* $\forall$ bzw. des *Existenzquantors* $\exists$ stehen. Ansonsten sind die Individuensymbole *frei*.

Definitionen

Term

es gilt:

1. die Wahrheitswerte w und f und alle Individuensymbole $x, \ldots$ sind Terme;
2. wenn f ein n-stelliges Funktionssymbol und $t_1, \ldots, t_n$ Terme sind, so ist auch $f(t_1, \ldots, t_n)$ ein Term;
3. andere Terme existieren nicht.

Elementarausdruck

es gilt:

1. wenn p ein n-stelliges Prädikatensymbol und $t_1, \ldots, t_n$ Terme sind, so ist auch $p(t_1, \ldots, t_n)$ ein elementarer Ausdruck;
2. wenn t_1 und t_2 Terme sind, so ist auch $t_1 = t_2$ ein elementarer Ausdruck.

prädikatenlogischer Ausdruck

es gilt:

1. jeder Elementarausdruck ist Ausdruck;
2. sind A und B Ausdrücke, so sind auch $\overline{A}, A \wedge B, A \vee B, A \rightarrow B$ und $A \leftrightarrow B$ Ausdrücke;
3. wenn $A(x)$ Ausdruck und x ein in $A(x)$ vorkommendes Individuensymbol ist, wobei in $A(x)$ keine Symbolfolge der Art $\forall x$ oder $\exists x$ vorkommt, so sind auch $\forall x A(x)$ oder $\exists x A(x)$ Ausdrücke;
4. nur die nach (1) bis (3) gebildeten Zeichenreihen sind prädikatenlogische Ausdrücke.

Zusammenhang zwischen Aussagen und prädikatenlogischen Ausdrücken

Prädikatenlogische Ausdrücke, in denen nur gebundene Individuensymbole vorkommen, stellen Aussagen dar.
Im folgenden werden diese als Mittel zur Definition genutzt.

Mengen

Mengendefinition	$\forall a(a \in A \leftrightarrow p_A(a))$ sprich: Für alle a gilt: a ist Element der Menge A genau dann, wenn $p_A(a)$ gilt (d.h. die Aussage *"Für a gilt $p_A(a)$"* ist *wahr*) oder $A = \{a \mid p_A(a)\}$ oder $A = \{a_0, a_1, \ldots, a_{m-1}\}$
Mengenmächtigkeit	$\lvert A\rvert = m \leftrightarrow A$ enthält m Elemente
speziell	$\lvert\emptyset\rvert = 0$ $\emptyset$: die leere Menge
disjunkte Menge	A disjunkt $B \leftrightarrow \overline{\exists a(a \in A \wedge a \in B)}$ $A \cap B = \emptyset$
Teilmenge	$A \subseteq B \leftrightarrow \forall a(a \in A \rightarrow a \in B) \vee A = \emptyset$
echte Teilmenge	$A \subset B \leftrightarrow \exists a(\overline{a \in A} \wedge a \in B) \wedge \forall a(a \in A \rightarrow a \in B)$
Mengengleichheit	$A = B \leftrightarrow \forall a(a \in A \leftrightarrow a \in B)$
$n-Tupel$	$C = [c_{n-1}, c_{n-2}, \ldots, c_0]$ (geordnete Menge)
Mengenprodukt (*Kreuzprodukt*)	$\forall a, b([a, b] \in A \times B \leftrightarrow a \in A \wedge b \in B)$ $A \times B$ (sprich: *"A kreuz B"*)
Mengenpotenz	$A^m = A^{m-1} \times A$ mit: $A^1 = A$
Potenzmenge	$\forall B(B \in \mathcal{P}(A) \leftrightarrow B \subseteq A)$
Mengenvereinigung	$\forall a(a \in A \cup B \leftrightarrow a \in A \vee a \in B)$
Mengenschnitt	$\forall a(a \in A \cap B \leftrightarrow a \in A \wedge a \in B)$
allgemein	$\bigcup_{i=1}^{m} A_i = \bigcup_{i=1}^{m-1} A_i \cup A_m$ bzw. $\bigcap_{i=1}^{m} A_i = \bigcap_{i=1}^{m-1} A_i \cap A_m$
Mengendifferenz	$\forall a(a \in B \setminus A \leftrightarrow a \in B \wedge \overline{a \in A})$
Komplement	$\mathcal{K}_M(A) = M \setminus A$ Komplement der Menge A bzgl. Grundmenge M
kurz	$\mathcal{K}_M(A) = \bar{A}$ bei allgemein bekannter Grundmenge
Partition	Für $\Pi(A)$ gilt: 1. $\lvert\Pi(A)\rvert > 1$ 2. $\forall M, N(M, N \in \Pi(A) \wedge M \neq N \rightarrow \overline{M = \emptyset} \wedge \overline{N = \emptyset} \wedge M \cap N = \emptyset)$ 3. $\forall a(a \in A \rightarrow \exists M(M \in \Pi(A) \wedge a \in M))$

Relationen

$n-stellige\ Relation$	$\mathcal{R} \subseteq A^n, \quad \forall t(t \in \mathcal{R} \leftrightarrow t \in A^n \wedge r(t))$
$Infixnotation$	$a\ r\ b$ sprich *"a ist in Relation r zu b"*

Eigenschaften zweistelliger Relationen $r \in \mathcal{R}$ mit $\mathcal{R} \subseteq A^2$; $\quad a, b, c \in A$

$Reflexivität$	$\forall a(a\ r\ a)$
$Irreflexivität$	$\forall a\overline{(a\ r\ a)}$
$Transitivität$	$\forall a, b, c(a\ r\ b \wedge b\ r\ c \rightarrow a\ r\ c)$
$Symmetrie$	$\forall a, b(a\ r\ b \rightarrow b\ r\ a)$
$Antisymmetrie$	$\forall a, b(a\ r\ b \wedge b\ r\ a \rightarrow a = b)$
$Asymmetrie$	$\forall a, b(a\ r\ b \rightarrow \overline{b\ r\ a})$
$Linearität$	$\forall a, b(a\ r\ b \vee b\ r\ a)$
$Konnexität$	$\forall a, b(a\ r\ b \vee a = b \vee b\ r\ a)$

Abbildungen

Abbildung $\mathcal{A}$	$\forall m, n([m, n] \in \mathcal{A} \leftrightarrow [m, n] \in M \times N \wedge p_A([m, n]))$
$Vorbereich(Vb)$	$\forall m(m \in Vb(\mathcal{A}) \leftrightarrow m \in M \wedge \exists n([m, n] \in \mathcal{A}))$
$Nachbereich(Nb)$	$\forall n(n \in Nb(\mathcal{A}) \leftrightarrow n \in N \wedge \exists m([m, n] \in \mathcal{A}))$
partielle Abbildung	$\mathcal{A}$ ist partiell $\leftrightarrow \exists m(m \in M \wedge \overline{m \in Vb(\mathcal{A})})$
$Funktion(\mathcal{F})$	eindeutige Abbildung, d.h. es gilt zusätzlich: $\forall m, n([m, n] \in \mathcal{F} \rightarrow \overline{\exists l(n \neq l \wedge [m, l] \in \mathcal{F})})$
symbolisch	$\mathcal{F} : M \Rightarrow N$ bzw. $n = \mathcal{F}(m)$ mit $n \in N$ und $m \in M$
$Operation(\mathcal{O})$	Funktion mit $Vb = M^n$ und $Nb = M$
symbolisch	$\mathcal{O} : M^n \Rightarrow M$ bzw. $m = \mathcal{O}[m_{n-1}, m_{n-2}, \ldots, m_0]$
$Kodierung(\mathcal{K})$	eindeutige Funktion, d.h. es gilt zusätzlich: $\forall m, n([m, n] \in \mathcal{K} \rightarrow \overline{\exists l(m \neq l \wedge [l, n] \in \mathcal{K})})$
symbolisch	$\mathcal{K} : M \Leftrightarrow N$ bzw. $n = \mathcal{K}(m)$ und $m = \mathcal{K}^{-1}(n)$
$Transformation(\mathcal{T})$	Kodierung mit $Vb = Nb = M$
symbolisch	$\mathcal{T} : M \Leftrightarrow M$

Literaturliste zur Lehrveranstaltung

"Technische Informatik – Teil RO"

H.-D. Wuttke; K. Henke	Schaltsysteme – Eine automatenorientierte Einführung, Pearson-Education Deutschland, München 2003
Th. Flick	Mikroprozessortechnik und Rechnerstrukturen, Springer Verlag, Berlin 2005
H.-J. Zander	Logischer Entwurf binärer Systeme, Verlag Technik, Berlin 1992
S. Hentschke	Grundzüge der Digitaltechnik, Teubner-Verlag, Stuttgart 1988
Informatik-Duden	Duden-Verlag, Mannheim, Wien, Zürich 2002
H.-D. Wuttke; K. Henke	Online-Materialien zur Lehrveranstaltung "Technische Informatik – Teil RO", TU Ilmenau, Fakultät IA, Ilmenau 2009, http://www.tu-ilmenau.de/ihs