Ostfalia
Hochschule für angewandte Wissenschaften

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Digitaltechnik Grundlagen

4. Schaltalgebra

Prof. Dr.-Ing. Thorsten Uelzen

Version 1.0 von 02/2018

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Gesetze und Rechenregeln

- Die **Schaltalgebra** ist die Anwendung der allgemeineren Booleschen Algebra auf Schaltungen aus binären Elementen (Schaltern, Relais, Transistoren im Schalterbetrieb).
- Andere Beispiele für Boolesche Algebren sind:
  - Aussagenlogik
  - Mengenlehre
  - Ereignisalgebra

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Gesetze und Rechenregeln

- Eine Algebra besteht aus einer Menge von *Elementen* und *Operationen* (Verknüpfungen, Funktionen) auf dieser Menge.
- Eine *Boolesche Algebra* besteht aus einer Menge B mit zwei Elementen: „0“ und „1“ → B = {0,1} und
- drei Operationen:
  - UND – „ $\wedge$ “
  - ODER – „ $\vee$ “
  - NICHT – „ $\overline{\ }$ “ oder „ $\neg$ “ oder „ / “ oder „ ! “ oder „ ´ “
- Die Elemente 0 und 1 von B nennt man *Boolesche Konstante*.
- Eine Variable x mit x $\in$ B nennt man *Boolesche Variable*.
- Die beiden *zweistelligen Operationen* „ $\wedge$“, „ $\vee$“ verknüpfen zwei Boolesche Variable miteinander zu einem Booleschen Ausdruck, der wiederum den Wert 0 oder 1 annehmen kann.
- Die einstellige Operation „ $\overline{\ }$ “ (nicht) gilt für eine Boolesche Variable.
- Enthält ein Boolescher Ausdruck mehrere Operationen, so können Klammern zur Festlegung der Auswertungsreihenfolge verwendet werden.

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Gesetze und Rechenregeln

Kommutativgesetze	(1)	$a \wedge b = b \wedge a$	(1')	$a \vee b = b \vee a$
Assoziativgesetze	(2)	$(a \wedge b) \wedge c = a \wedge (b \wedge c)$	(2')	$(a \vee b) \vee c = a \vee (b \vee c)$
Idempotenzgesetze	(3)	$a \wedge a = a$	(3')	$a \vee a = a$
Distributivgesetze	(4)	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	(4')	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
Neutralitätsgesetze	(5)	$a \wedge 1 = a$	(5')	$a \vee 0 = a$
Extremalgesetze	(6)	$a \wedge 0 = 0$	(6')	$a \vee 1 = 1$
Doppelnegationsgesetz (Involution)	(7)	$\neg(\neg a) = a$		
De Morgansche Gesetze	(8)	$\neg(a \wedge b) = \neg a \vee \neg b$	(8')	$\neg(a \vee b) = \neg a \wedge \neg b$
Komplementärgesetze	(9)	$a \wedge \neg a = 0$	(9')	$a \vee \neg a = 1$
Dualitätsgesetze	(10)	$\neg 0 = 1$	(10')	$\neg 1 = 0$
Absorptionsgesetze	(11)	$a \vee (a \wedge b) = a$	(11')	$a \wedge (a \vee b) = a$

Vergleich Arbeitsblätter 4.1

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Gesetze und Rechenregeln

- Die Idee der **Schaltalgebra** ist die **Anwendung der Booleschen Algebra** mit Hilfe von Schaltern und Relais.
- Sie hat bis heute nichts an Aktualität verloren und ist die **Grundlage der Digitaltechnik**
- Eine n-stellige, m-wertige Schaltfunktion f ordnet n unabhängigen binären Schaltvariablen m abhängige Schaltvariable (den Funktionswert) zu; sie ist also eine **Abbildung $\{0;1\}^n \rightarrow \{0;1\}^m$** .

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Gesetze und Rechenregeln

Funktion einer Schaltvariablen

- f(a) = 0 a ist immer „0“ – macht wenig Sinn
- f(a) = 1 a ist immer „1“ – macht auch wenig Sinn
- f(a) = a Identität
- f(a) = ¬a Negation

Funktionen von zwei bzw. n Schaltvariablen

Vergleich Arbeitsblätter 4.1

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Gesetze und Rechenregeln

Funktionen von zwei bzw. n Schaltvariablen

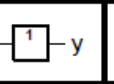
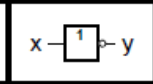
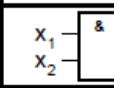
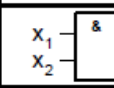
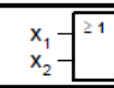
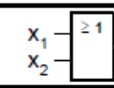
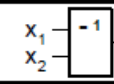
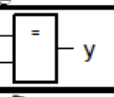
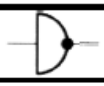
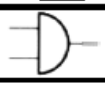
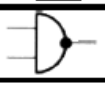
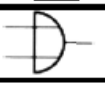
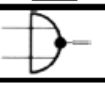
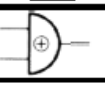
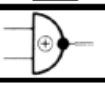
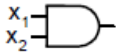
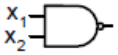
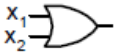
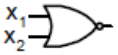
Name	Schreibweisen			Sprech-weise	Wertetabelle			
					$a=0$		$a=1$	
	Nach DIN 66000	In der VL zusätzlich verwendet	sonstige		$b=0$	$b=1$	$b=0$	$b=1$
UND, AND, Konjunktion	$a \wedge b$	ab	$a*b$, $a \& b$, $a \cdot b$	a und b	0	0	0	1
NAND	$a \overline{\wedge} b$	$\overline{a \wedge b}$, $\overline{ab}$	$\overline{a*b}$, $\overline{a \& b}$, $\overline{a \cdot b}$	a nand b	1	1	1	0
ODER, OR, Disjunktion	$a \vee b$		$a+b$	a oder b	0	1	1	1
NOR	$a \overline{\vee} b$	$\overline{a \vee b}$	$\overline{a+b}$	a nor b	1	0	0	0
XOR, Anti-valenz, Ex-klusiv-ODER	$a \nleftrightarrow b$		$a \not\equiv b$, $a \oplus b$, $a \veebar b$, $a \neq b$	a xor b	0	1	1	0
Äquivalenz	$a \leftrightarrow b$		$a \equiv b$, $a \otimes b$, $\overline{a \veebar b}$, $a \sim b$	a äqui-valent b	1	0	0	1

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Gesetze und Rechenregeln

Funktionen von zwei bzw. n Schaltvariablen

Name	Identität, Puffer	NICHT, Negation	UND, Konjunktion	NAND	ODER, Disjunktion	NOR	XOR, Antivalenz	Äquivalenz
Schaltfunktion nach DIN 66000	$y = x$	$y = \overline{x}$, $y = \neg x$	$y = x_1 \wedge x_2$	$y = \overline{x_1 \wedge x_2}$	$y = x_1 \vee x_2$	$y = \overline{x_1 \vee x_2}$	$y = x_1 \nleftrightarrow x_2$	$y = x_1 \leftrightarrow x_2$
Wahrheitstabelle	$x \mid y$ 0 | 0 1 | 1	$x \mid y$ 0 | 1 1 | 0	$x_1\ x_2 \mid y$ 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1	$x_1\ x_2 \mid y$ 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	$x_1\ x_2 \mid y$ 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1	$x_1\ x_2 \mid y$ 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	$x_1\ x_2 \mid y$ 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	$x_1\ x_2 \mid y$ 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1
Kurzbeschreibung			1 g. d., w.[1] alle Eingänge 1	0 g. d., w. alle Eingänge 1	0 g. d., w. kein Eingang 1	1 g. d., w. kein Eingang 1	1 g. d., w. genau ein Eingang 1	1 g. d., w. alle Eingänge gleich
Schaltsymbol nach DIN EN 60617-12								
Schaltsymbol nach alter DIN 40700-14								
Schaltsymbol nach IEEE Std 91-1984								
VHDL-Code	y <= x;	y <= not x;	y <= x1 and x2;	y <= x1 nand x2;	y <= x1 or x2;	y <= x1 nor x2;	y <= x1 xor x2;	y <= x1 xnor x2;

Vergleich Arbeitsblätter 4.1

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Gesetze und Rechenregeln

- **Jede beliebige Verknüpfung** von beliebig vielen Schaltvariablen kann beschrieben werden unter
  - ausschließlicher Verwendung der Grundverknüpfungen **UND, ODER, NICHT**
  - oder ausschließlicher Verwendung der **NAND-Verknüpfung**
  - oder ausschließlicher Verwendung der **NOR-Verknüpfung**.
- Früher waren ICs, die 4 NAND-Verknüpfungen mit je 2 Eingängen enthielten (SN 7400) besonders günstig. Daher wurden manche komplexen Logik-Schaltungen fast nur aus diesen ICs aufgebaut.
- Aktuelle Anwendung: Realisierung von Logik mittels konfigurierbarer ICs (programmable logic devices, PLDs) wie GALs (Generic Array Logic) und CPLDs (Complex Programmable Logic Device), die Felder von NAND-/NOR-Verknüpfungen enthalten.

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Schaltfunktion

- **Formalismus** aus der mathematischen Logik bekannt.
- Wird **hauptsächlich in der Lehre** verwendet.
- In der Praxis unüblich, da bei vielen Variablen sehr schlecht erfassbar und unanschaulich und maschinell schlecht verarbeitbar.
  → Besser: Grafische Darstellung mittels Symbolen
- Hat man mehr als 2 Eingangsvariablen bzw. auch Ausgangsvariablen, so lassen sich diese zurückführen auf einwertige Schaltfunktionen
- Die zur **Formulierung von Schaltfunktionen zu verwendenden Symbole** sind in DIN 66000 festgelegt
  (siehe **Arbeitsblatt** „4.1 Wichtige Schaltfunktionen“, Tabelle 1, 2. Spalte).

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Schaltfunktion

- Wichtig ist die Auswertereihenfolge, die auch in der DIN festgelegt ist. Um sicher zu gehen, sollten jedoch bei der Formulierung Klammern gesetzt werden.

Auswertungsreihenfolge (Priorität der Operatoren) nach DIN:

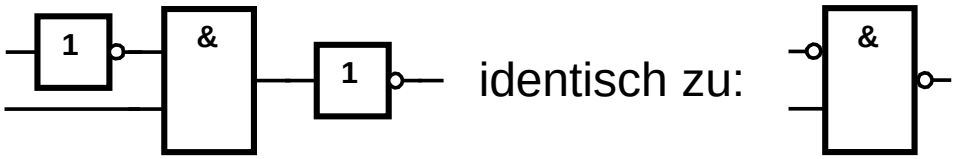
Höchste:	$\ldots, \neg \ldots$ (NICHT)
Mittlere:	$\wedge, \vee, \overline{\wedge}, \overline{\vee}$ (UND, ODER, NAND, NOR)
Niedrigste:	$\leftrightarrow, \not\leftrightarrow$ (Äquivalenz, Antivalenz)

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Schaltsymbol

- Die Nutzung von Schaltfunktionen kann zu sehr unübersichtlichen Verknüpfungen führen → **besser Schaltsymbole**
- Die Beschreibung logischer Schaltungen mittels **Schaltbildern kann sehr übersichtlich und schnell erfassbar** sein und wird daher in der **Praxis** sehr häufig eingesetzt.
- An Stelle eines separaten Negationsgatters darf rein logisch immer auch ein Negationskreis an einem anderen Gatter verwendet werden. Bzgl. der technischen Realisierung gibt es aber oft einen Unterschied!
- **Beispiel:**



Vergleich Arbeitsblätter 4.2

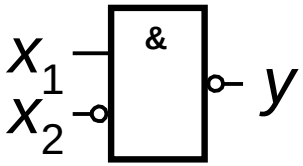
[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

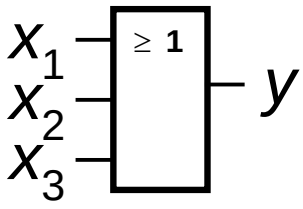
Schaltsymbol

- **„Stürzen“ von Gattern:**
  Ein Negationspunkt am Ausgang eines UND- oder ODER-Gatters darf zu allen Eingängen verschoben werden. Dabei wird aus einem UND-Gatter ein ODER-Gatter und umgekehrt.
- **Übungsaufgaben an der Tafel:**
  Stürzen Sie die folgenden Gatter und vollziehen Sie die Ergebnisse mittels der Gesetze von de Morgan nach:

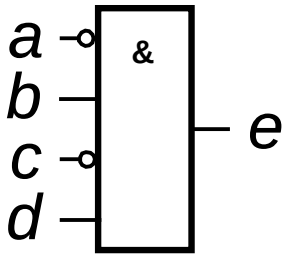
a)



b)



c)



→ **siehe Tafel**

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Schaltsymbol

Aufgaben zum Üben:

Realisieren Sie unter ausschließlicher Verwendung von
a) UND-, ODER-, NICHT-Gattern
die Funktionen NAND, NOR, XOR
b) NAND-Gattern
die Funktionen UND, ODER, NICHT, NOR, XOR
c) NOR-Gattern
die Funktionen UND, ODER, NICHT, NAND, XOR.

Alle Gatter außer NICHT sollen 2 Eingänge haben.
Leiten Sie die Umformungen mittels der Regeln der Schaltalgebra her!

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Wahrheitstabelle

- Wahrheitstabellen helfen in der **Lehre**, um Funktionen zu verstehen und zu verifizieren.
- In der **Praxis haben Sie keine Bedeutung**, da sie viel zu groß wären. (Überlege: 8 Bit E/A)
- Wahrheitstabellen sind **immer zweigeteilt**: Eingänge und Ausgänge
- Die Eingänge müssen alle möglichen Schaltzustände abbilden; bei n **Eingängen sind die 2^n**
- **Beispiele:**

Tabelle 1 Wahrheitstabelle einer zweistelligen, zweiwertigen Schaltfunktion

x_1	x_0	y_1	y_2
0	0	1	X
0	1	0	1
1	0	0	1
1	1	1	0

Don‘t care Symbol im Ausgang: einfachere Schaltungstechnische Realisierung

Tabelle 2 Wahrheitstabelle mit don’t care-Symbolen auf der linken Seite

a	b	c	e	f
0	X	X	1	0
1	0	0	1	0
1	1	0	0	1
1	X	1	1	1

Don‘t care Symbole im Eingang: kompakteren Schreibweise

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Karnaugh-Veitch-Diagramm (KV-Diagramm)

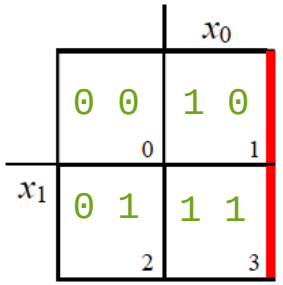
- **auch:** KV-Tafel, Karnaugh-Diagramm, Karnaugh-Plan, K-Plan, Symmetriediagramm; englisch: K-map
- Verwendung **zur grafischen Darstellung und Vereinfachung** (Minimierung) von einwertigen Schaltfunktionen mit n = 2 ... 6 (praktisch meist nur bis n = 4) Eingangsvariablen
- Ein KV-Diagramm für n Eingangsvariablen hat **2^n Felder** (siehe Beispiele).
- Das KV-Diagramm wird mit den **Variablen an den Rändern** beschriftet. Dabei kommt jede Variable in negierter und nicht-negierter Form vor.
- Die Zuordnung der Variablen zu den einzelnen Feldern kann dabei beliebig erfolgen, jedoch ist zu beachten, dass sich **horizontal und vertikal benachbarte Felder nur in genau einer Variablen unterscheiden** dürfen (vgl. Gray-Code).
- In jedes der **Felder wird der Wert der Schaltfunktion** (0 oder 1) für die zugehörige Kombination der Werte der n Eingangsvariablen eingetragen. Felder, in denen der Wert der Schaltfunktion nicht vorgegeben, also beliebig wählbar ist, werden mit don’t care-Symbolen wie „X“ oder „d“ gekennzeichnet.

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Karnaugh-Veitch-Diagramm (KV-Diagramm)

Beispiele:



- entspricht

Bild 1 KV-Diagramm für 2 Variablen

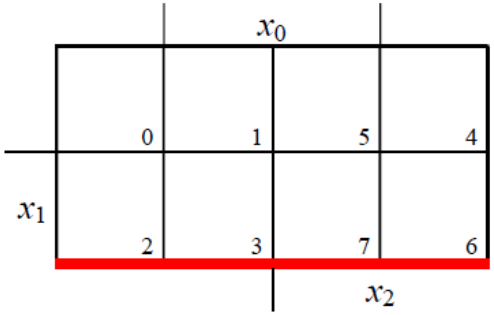


Bild 2 KV-Diagramm für 3 Variablen

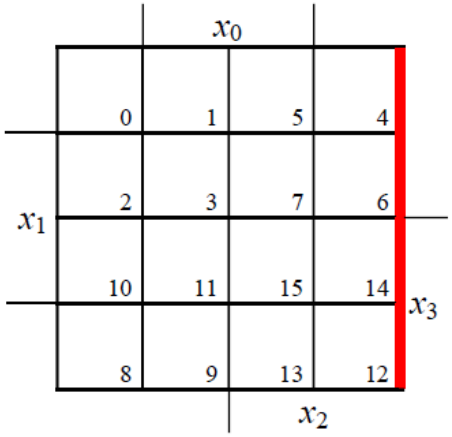


Bild 3 KV-Diagramm für 4 Variablen

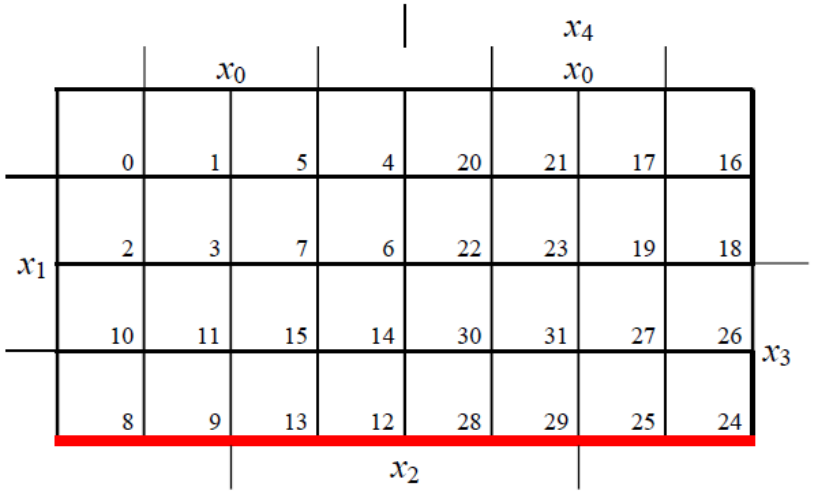


Bild 4 KV-Diagramm für 5 Variablen

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Karnaugh-Veitch-Diagramm (KV-Diagramm)

Übungsaufgabe:

Gegeben:
Schaltfunktion $y = f(x_0, x_1, x_2, x_3)$, die genau dann den Wert 1 liefert, wenn die Tetrade $x_3\ x_2\ x_1\ x_0$ (also x_3 MSB und x_0 LSB)
als Dualzahl interpretiert einen Wert größer 9_{10} hat.

Gesucht:
a) Wahrheitstabelle für y
b) KV-Diagramm für y (eintragen in Bild 3 des AB)

→ Siehe Tafel

Was bringt das jetzt? … kommt noch!

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

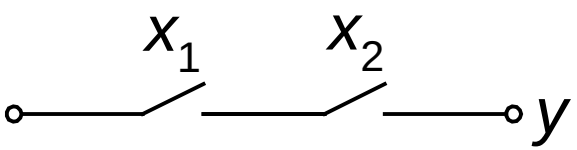
Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Netzwerk aus Schaltern

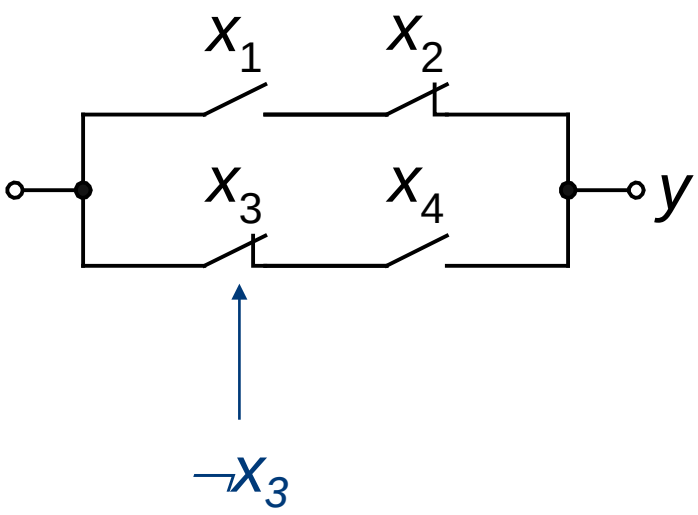
- **UND-Verknüpfung: Reihenschaltung**
- **ODER-Verknüpfung: Parallelschaltung**
- Für kompliziertere Funktionen benötigt man Relais, die von Teilfunktionen angesteuert werden.

- **Übungsaufgabe:**
  Welche Schaltfunktionen $y = f(x_\mu)$ werden durch folgende Netzwerke aus Schaltern repräsentiert?

a)



b)



→ **Siehe Tafel**

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Impulsdiagramm

- Ein Impulsdiagramm oder Zeitablaufdiagramm dient zur **Visualisierung der Funktionsweise digitaler Schaltungen**.
- Es stellt die Veränderung der Ausgangssignale bei Variation der Eingangssignale dar.
- Entspricht der **grafischen Darstellung des Inhalts einer Wahrheitstabelle**. Die einzelnen Zeilen der WT entsprechen hier Abschnitten auf der (gleichmäßig geteilten) Zeitachse.
- Wird in der **Praxis häufig in Datenblättern** verwendet.
- Angegeben werden entweder logische Werte (0, 1) oder Pegel (low = L, high = H), die meist elektrischen Potenzialen entsprechen.
- **Beispiel: TTL-Schaltungen**

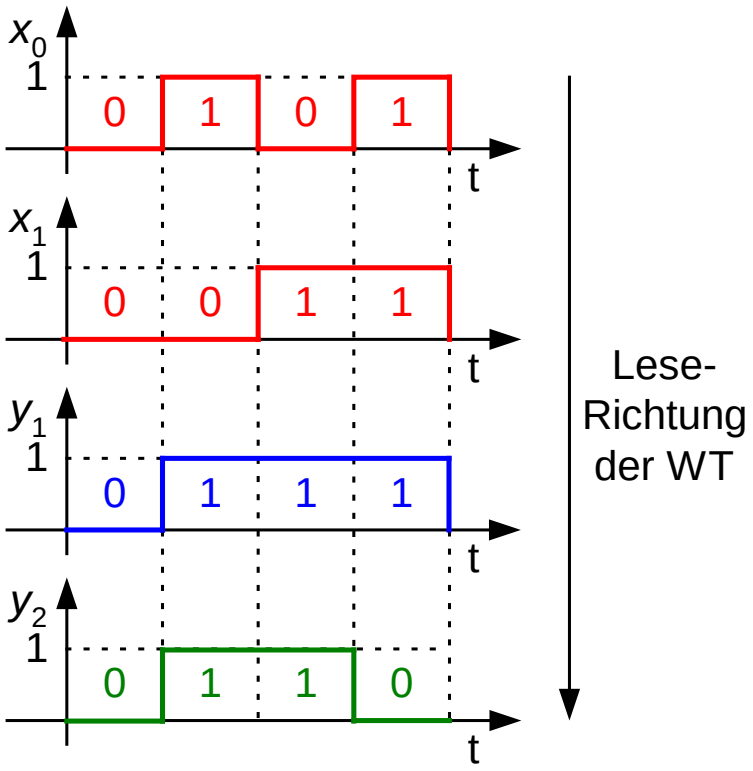
  0 … 0,4 V → L
  2,4 … 5 V → H

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Impulsdiagramm

- **Beispiel siehe Abbildung**
- Welche Funktionen y_1 und y_2 sind zu erkennen?

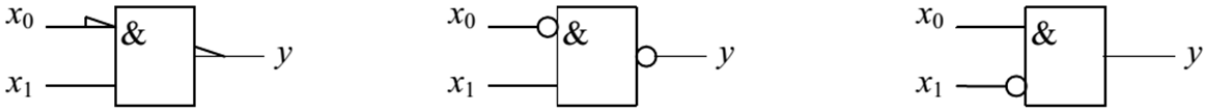


[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Impulsdiagramm

- **Zuordnung zwischen Pegeln und logischen Werten frei wählbar**:
  - Positive Logik (active high): $\mathrm{L} \mathrel{\hat{=}} 0; \quad \mathrm{H} \mathrel{\hat{=}} 1$
  - Negative Logik (active low): $\mathrm{L} \mathrel{\hat{=}} 1; \quad \mathrm{H} \mathrel{\hat{=}} 0$
- Bei der Verwendung von Pegeln wird an Schaltsymbolen zur Kennzeichnung einer Negation ein halber Pfeil statt eines Kreises verwendet.



x_0	x_1	y
L	L	H
L	H	L
H	L	H
H	H	H

a)

x_0	x_1	y
0	0	1
0	1	0
1	0	1
1	1	1

b)

x_0	x_1	y
1	1	0
1	0	1
0	1	0
0	0	0

c)

Bild 4-2 Schaltsymbole und Wahrheitstabellen für a) Pegeldarstellung b) positive Logik c) negative Logik.

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Hardware-Beschreibungssprachen

- In der Praxis am weitesten verbreitete Beschreibungsform von Schaltalgebra.
- Bekannteste Vertreter: **VHDL, Verilog-HDL, SystemC**
- Sie können in der **VL Design Digitaler Systeme** die Grundlagen von VHDL lernen.
- Haben sehr große Ähnlichkeit mit „normalen“ Programmiersprachen wie C, C++, …
- Rechnergestützter Entwurf elektronischer Schaltungen und Systeme: Electronic Design Automation (EDA).
  Ist eine Anwendung des Computer Aided Design (CAD).
- Prinzipieller Ablauf des Entwurfs komplexer digitaler integrierter Schaltungen

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Hardware-Beschreibungssprachen

Beispiel für VHDL

Phase 1: ASIC-Spezifikation

Der Vorwärts/Rückwärts-Zähler soll folgendermaßen entwickelt werden.

4 Bit Zähler mit den Datenausgängen C0, C1, C2, C3.

- Dieser zählt von 0 bis 15 bzw. in entgegengesetzter Richtung.
- Reset Eingang, setzt bei Reset=0 den Zähler auf den Wert 0 zurück
- Clock Eingang; mit jedem Wechsel von Clock von 0 nach 1 zählt der Zähler
- UpDown Eingang zur Festlegung der Zählrichtung; 1=Up; 0=Down
- Count Eingang, ist Count=1 wird mit jedem 0-1 Wechsel von Clock Wert des Zählers um eins erhöht (Up), bzw. um eins erniedrigt (Down). Bei Count = 0 bleibt
  der alte Zählerstand erhalten
- Bei einem Zählerüber- oder -unterlauf soll wieder bei 0 bzw. 15 weitergezählt werden

[Quelle: http://www.andreas-schwope.de]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Hardware-Beschreibungssprachen

Beispiel für VHDL

Phase 2: ASIC-Kodierung

Der folgende Abschnitt in kursiv stellt den vollständigen VHDL Code des Zählers dar, der oben spezifiziert worden ist. Um diesen auf seine vollständige und korrekte Implementierung zu prüfen, sind Simulationen notwendig.

```
library IEEE;
use IEEE.std_logic_1164.all; -- import std_logic types
use IEEE.std_logic_arith.all; -- import add/sub of std_logic_vector

ENTITY counter_updown is
PORT ( -- Counter Control Signals
       clock           : IN  std_logic;  -- counter clock
       reset           : IN  std_logic;  -- counter reset
       updown          : IN  std_logic;  -- count direction up/down
       count           : IN  std_logic;  -- count enable
       -- Counter Output Signals
       c0, c1, c2, c3  : OUT std_logic); -- counter bit 0..3
END counter_updown;

ARCHITECTURE rtl OF counter_updown IS
  -- internal signal declaration
  SIGNAL counter : std_logic_vector(3 DOWNTO 0);
  BEGIN

   count_process : PROCESS
   -- This process builds the counter
   BEGIN
   -- wait for rising edge of clock (0->1)
   WAIT UNTIL clock'EVENT AND clock = '1';
   IF reset = '0' THEN
     -- set all counter bits to zero
     counter <= (others => '0');
ELSE
     IF (count = '1') AND (updown = '1') THEN
       -- count up: n, n+1, n+2, ..., 15, 0, 1, ...
       counter <= CONV_STD_LOGIC_VECTOR(UNSIGNED(counter) + 1, 4);
     ELSIF (count = '1') AND (updown = '0') THEN
       -- count down: n, n-1, n-2, ..., 0, 15, 14, ...
       counter <= CONV_STD_LOGIC_VECTOR(UNSIGNED(counter) - 1, 4);
     ELSE
       -- hold counter in the same value as before
       counter <= counter;
     END IF;
   END IF;
   END PROCESS;

   -- Connect internal counter to outputs
   c0 <= counter(0);
   c1 <= counter(1);
   c2 <= counter(2);
   c3 <= counter(3);

  END rtl;
```


[Quelle: http://www.andreas-schwope.de]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Hardware-Beschreibungssprachen

Beispiel für VHDL

Phase 3: Testbench

Der nachfolgende Code ist die VHDL Testbench unseres Zählers. Mit einer Testbench wird die Funktionalität des Designs in allen Bereichen überprüft.

```
library IEEE;
use IEEE.std_logic_1164.all;

ENTITY testbench IS
END testbench;

ARCHITECTURE beh OF testbench IS

COMPONENT counter_updown
-- component interface of 4 bit up/down counter
PORT ( -- Counter Control Signals
       clock   : IN  std_logic;  -- counter clock
       reset   : IN  std_logic;  -- counter reset
       updown  : IN  std_logic;  -- count direction up/down
       count   : IN  std_logic;  -- count enable
       -- Counter Output Signals
       c0      : OUT std_logic;  -- counter bit 0
       c1      : OUT std_logic;  -- counter bit 1
       c2      : OUT std_logic;  -- counter bit 2
       c3      : OUT std_logic); -- counter bit 3
end COMPONENT;

-- define all testbench signals (same names as counter)
signal c0, c1, c2, c3              : std_logic;
signal clock, reset, updown, count : std_logic;

BEGIN

  ctr  : counter_updown
  -- connect 4 bit up/down conter with testbench signals
  PORT MAP (clock, reset, updown, count, c0, c1, c2, c3);

ctr_proc : PROCESS
  -- generate control signals for different count sequences
  VARIABLE i : INTEGER;
  BEGIN
   WAIT UNTIL clock'EVENT AND clock = '1';
   reset  <= '0';
   WAIT UNTIL clock'EVENT AND clock = '1';
   reset  <= '1';
   count  <= '1'; -- start counting
   updown <= '1'; -- count up
   FOR i IN 1 TO 18 LOOP
     WAIT UNTIL clock'EVENT AND clock = '1';
   END LOOP;
   count  <= '0'; -- stop counting
   WAIT UNTIL clock'EVENT AND clock = '1';
   updown <= '0'; -- count down
   count  <= '1'; -- start counting
   FOR i IN 1 TO 10 LOOP
     WAIT UNTIL clock'EVENT AND clock = '1';
   END LOOP;
   -- process continues endlessly with first line
  END PROCESS ctr_proc;

  clk_proc : PROCESS (clock)
  -- generate counter clock
  -- periode is 200ns / 5 MHz
  BEGIN
   IF clock'EVENT AND clock = '1' then
     clock <= '0' AFTER 100 NS;
   ELSE
     clock <= '1' AFTER 100 NS;
   END IF;
  END PROCESS clk_proc;

END beh;
```

[Quelle: http://www.andreas-schwope.de]

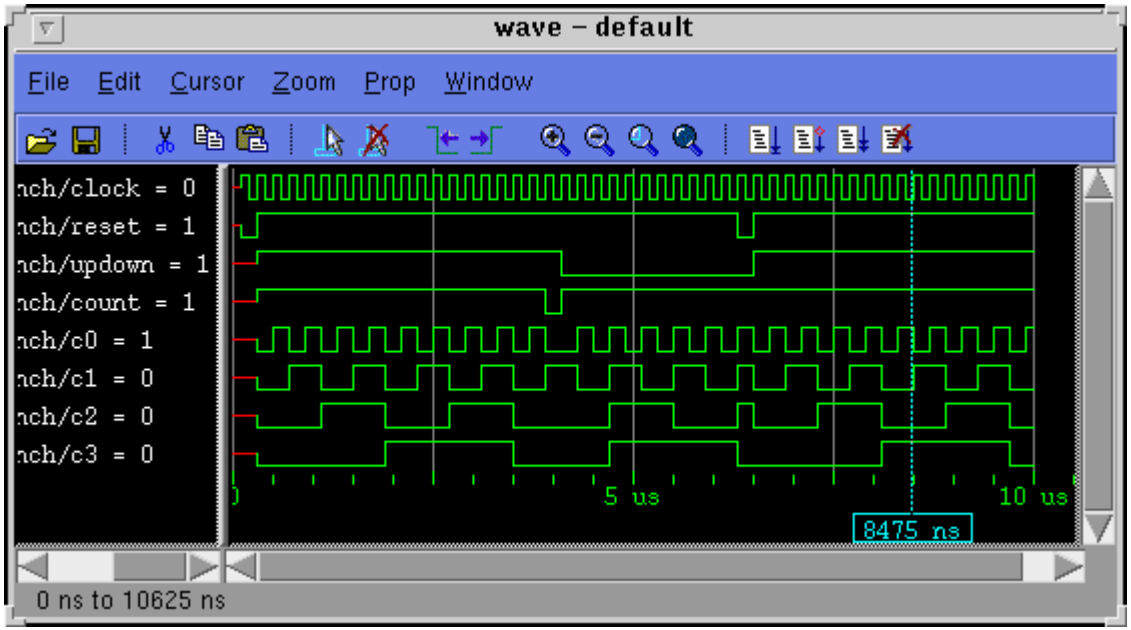
Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Hardware-Beschreibungssprachen

Beispiel für VHDL

Phase 4: Simulation

Die Ergebnisse der Testbenchsimulation werden in einem Waveform Viewer dargestellt.



Freeware für VHDL auf Linux-Basis:

www-asim.lip6.fr/recherche/alliance

www.freehdl.seul.org

www.ece.uc.edu/~paw/savant

[Quelle: http://www.andreas-schwope.de]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Hardware-Beschreibungssprachen

Beispiel für VHDL

Phase 6: ASIC-Layout

Das ASIC-Layout wird in zwei Stufen durchgeführt. In der ersten Stufe werden zunächst alle Schaltungselemente der Netzliste auf der Chipfläche, dem sogenannten Die angeordnet.

Core-Area

Dieser Bereich befindet sich in der Mitte des Dies (grau). In diesem wird die komplette Netzliste ohne die Eingangs- und Ausgangs-Buffer angeordnet.

IO- und Power-Ring

Dieser Bereich (gelb/grün) beinhaltet die elektrische Stromversorgung (Power-Supply) der Eingangs- und Ausgangsbuffer auf dem IO-Ring und der umschlossenen Core-Area.

Pad-Area

Dieses ist der äußerste Bereich des Dies außerhalb des Power- und IO-Rings. In diesem befinden sich relativ große metallisierte Flächen (gelbe Flächen im roten Randbereich), Pads genannt, über die der Die mit dem Gehäuse verbunden wird (Bonding).

[Quelle: http://www.andreas-schwope.de]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Hardware-Beschreibungssprachen

Beispiel für VHDL

Phase 6: ASIC-Produktion

Eine ausführliche Darstellung des Produktionsprozesses eines ASICs würde den Rahmen hier sprengen. Es handelt sich jedoch um aufwändige und komplizierte Prozesse in Reinraumumgebung.

Selbststudium Arbeitsblätter 4.2.7

[Quelle: http://www.andreas-schwope.de]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Normalformen

- Normalformen =
  beliebige **Schaltfunktionen in bestimmter, einheitlicher Form**
- Verwendung von **bestimmten, einfachen Strukturen von Gattern**
- **Wichtigste** Normalformen sind:
  - **disjunktive Normalformen** (Disjunktion = ODER)
  - **konjunktive Normalformen** (Konjunktion = UND)
- **Beliebig viele Eingangsvariablen** (Anzahl n = binäre Eingangsvariable) ergeben **eine Ausgangsvariable** unter alleiniger Verwendung von NICHT-, UND- und ODER-Verknüpfungen
- Schaltnetz = **zweistufig** (ohne NICHT)
- Jede binäre Eingangsvariable wird als Dualzahl interpretiert

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Normalformen - Minterm

- Minterm → auch: **Vollkunjunktion**
- Ein Minterm ist eine **UND-Verknüpfung** aller (ggf. negierten) Eingangsvariablen einer Schaltfunktion.
- Ein Minterm ergibt nur bei der **minimal** sinnvollen Anzahl von Verknüpfungen der Eingangsvariablen eine **1**, nämlich **einer**!
- Es gibt 2^n Minterme.
- In einem Minterm sind die zu diesem Term gehörigen **Eingangsvariablen**, die den Wert **0** haben, **invertiert.**

Beispiel:

Eingangsvariable:	$x = 101_2$	$= 5_{10}$
Minterm:	$m_5 = x_2 \wedge \neg x_1 \wedge x_0$	

m_5 wird nur bei genau dieser Eingangsvariablen „1“

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Normalformen - Maxterm

- Maxterm → auch: **Volldisjunktion**
- Ein Maxterm ist eine **ODER-Verknüpfung** aller (ggf. negierten) Eingangsvariablen einer Schaltfunktion.
- Ein Maxterm ergibt bei allen Werten des Eingangsvektors **außer einem** den Wert **1**, also **nur in einem einzigen Fall eine 0**.

- In einem Maxterm sind die zu diesem Term gehörigen **Eingangsvariablen**, die den Wert **1** haben, **invertiert.**

Beispiel:

Eingangsvariable:	$x = 101_2$	$= 5_{10}$
Maxterm:	$M_5 = \neg x_2 \vee x_1 \vee \neg x_0$	

M_5 wird nur bei genau dieser Eingangsvariablen „0“

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Normalformen – alle Minterme und Maxterme für 3 Bit

Eingangsvektor			Eingangsvektor als Dualzahl	zugehöriger **Minterm**	zugehöriger **Maxterm**	Ausgangsvariable		
x_2	x_1	x_0				y_a	y_b	y_c
0	0	0	0	$m_0 = \overline{x_2}\,\overline{x_1}\,\overline{x_0}$	$M_0 = x_2 \vee x_1 \vee x_0$	0	1	1
0	0	1	1	$m_1 = \overline{x_2}\,\overline{x_1}x_0$	$M_1 = x_2 \vee x_1 \vee \overline{x_0}$	0	0	0
0	1	0	2	$m_2 = \overline{x_2}x_1\overline{x_0}$	$M_2 = x_2 \vee \overline{x_1} \vee x_0$	1	1	0
0	1	1	3	$m_3 = \overline{x_2}x_1x_0$	$M_3 = x_2 \vee \overline{x_1} \vee \overline{x_0}$	1	1	0
1	0	0	4	$m_4 = x_2\,\overline{x_1}\,\overline{x_0}$	$M_4 = \overline{x_2} \vee x_1 \vee x_0$	0	1	0
1	0	1	5	$m_5 = x_2\,\overline{x_1}x_0$	$M_5 = \overline{x_2} \vee x_1 \vee \overline{x_0}$	0	1	1
1	1	0	6	$m_6 = x_2x_1\overline{x_0}$	$M_6 = \overline{x_2} \vee \overline{x_1} \vee x_0$	0	1	1
1	1	1	7	$m_7 = x_2x_1x_0$	$M_7 = \overline{x_2} \vee \overline{x_1} \vee \overline{x_0}$	0	0	1

… kommt zwei Folien weiter

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Disjunktive Normalform - DNF

- Eine **disjunktive Normalform** (DNF) ist eine **ODER-Verknüpfung von UND-Verknüpfungen**, die eine bestimmte Schaltfunktion realisiert.
- **Annahme:** alle Eingangsvariablen sind in normaler und negierter Form verfügbar.
- Zu einer Schaltfunktion existieren i. a. **mehrere verschiedene DNFs**.
- Eine DNF wird i. a. durch ein **zweistufiges Schaltnetz** (Negierer werden nicht mitgezählt) realisiert.

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Disjunktive Normalform – DNF
Kanonische disjunktive Normalform - KDNF

- Die kanonische disjunktive Normalform (KDNF) einer Schaltfunktion ist die **Disjunktion** (ODER-Verknüpfung) derjenigen **Minterme** der Funktion, für die die Schaltfunktion den Wert 1 liefert. Zu jeder Schaltfunktion gibt es **genau eine KDNF**.

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Disjunktive Normalform – DNF
Kanonische disjunktive Normalform - KDNF

Eingangsvektor			Eingangsvektor als Dualzahl	zugehöriger **Minterm**	zugehöriger **Maxterm**	Ausgangsvariable		
x_2	x_1	x_0				y_a	y_b	y_c
0	0	0	0	$m_0 = \overline{x_2}\,\overline{x_1}\,\overline{x_0}$	$M_0 = x_2 \vee x_1 \vee x_0$	0	1	1
0	0	1	1	$m_1 = \overline{x_2}\,\overline{x_1}x_0$	$M_1 = x_2 \vee x_1 \vee \overline{x_0}$	0	0	0
0	1	0	2	$m_2 = \overline{x_2}x_1\overline{x_0}$	$M_2 = x_2 \vee \overline{x_1} \vee x_0$	1	1	0
0	1	1	3	$m_3 = \overline{x_2}x_1x_0$	$M_3 = x_2 \vee \overline{x_1} \vee \overline{x_0}$	1	1	0
1	0	0	4	$m_4 = x_2\,\overline{x_1}\,\overline{x_0}$	$M_4 = \overline{x_2} \vee x_1 \vee x_0$	0	1	0
1	0	1	5	$m_5 = x_2\,\overline{x_1}x_0$	$M_5 = \overline{x_2} \vee x_1 \vee \overline{x_0}$	0	1	1
1	1	0	6	$m_6 = x_2x_1\overline{x_0}$	$M_6 = \overline{x_2} \vee \overline{x_1} \vee x_0$	0	1	1
1	1	1	7	$m_7 = x_2x_1x_0$	$M_7 = \overline{x_2} \vee \overline{x_1} \vee \overline{x_0}$	0	0	1

$$y_a = (\overline{x_2}x_1\overline{x_0}) \vee (\overline{x_2}x_1x_0) = m_2 \vee m_3$$

$$y_b = (\overline{x_2}\,\overline{x_1}\,\overline{x_0}) \vee (\overline{x_2}x_1\overline{x_0}) \vee (\overline{x_2}x_1x_0) \vee (x_2\,\overline{x_1}\,\overline{x_0}) \vee (x_2\,\overline{x_1}x_0) \vee (x_2x_1\overline{x_0}) = m_0 \vee m_2 \vee m_3 \vee m_4 \vee m_5 \vee m_6$$

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Disjunktive Normalform – DNF
(Allgemeine) disjunktive Normalform - DNF

- engl. auch: **SOP – Sum of Products**
- Die (allgemeine) disjunktive Normalform (DNF) einer Schaltfunktion ist die **Disjunktion** (**ODER-Verknüpfung**) und die **UND-Verknüpfungen** von Eingangsvariablen (in normaler oder negierter Form).
- Eine **DNF** enthält:
  - zwischen 1 und 2^n UND-Verknüpfungen von jeweils 2 bis n Eingangsvariablen
  - sowie i. a. eine ODER-Verknüpfung der Ergebnisse der UND-Verknüpfungen.
- Zu einer Schaltfunktion gibt es neben der KDNF meist weitere Darstellungen in DNF

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Disjunktive Normalform – DNF

Übungsaufgabe:

Schaltnetz, das die Funktion y_a in KDNF realisiert.

Alle Eingangsvariablen werden der Übersichtlichkeit halber negiert und dann von den 6 Signalen die benötigten an die UND-Gatter der 1. Stufe gelegt. „Bus-artige“ Darstellung vorteilhaft.

Die Ausgänge aller UND-Gatter werden an die Eingänge des einen ODER-Gatters der 2. Stufe gelegt.

→ Siehe Tafel

Weitere:

Funktionen y_b, y_c in KDNF.

[Quelle: Fricke, K.: Digitaltechnik, Springer Vieweg und Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Konjunktive Normalformen

- Eine **konjunktive Normalform** (KNF) ist eine **UND-Verknüpfung** von **ODER-Verknüpfungen**, die eine bestimmte Schaltfunktion realisiert.
- **Annahme:** alle Eingangsvariablen sind in normaler und negierter Form verfügbar.
- Zu einer Schaltfunktion existieren i. a. **mehrere verschiedene KNFs**.
- Eine KNF wird i. a. durch ein **zweistufiges Schaltnetz** (Negierer werden nicht mitgezählt) realisiert.

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Konjunktive Normalform – DNF
Kanonische konjunktive Normalform - KKNF

- Die **kanonische konjunktive Normalform (KKNF)** einer Schaltfunktion ist die **Konjunktion (UND-Verknüpfung)** derjenigen **Maxterme** der Funktion, für die die Schaltfunktion den Wert „0“ liefert.
- Zu jeder Schaltfunktion gibt es genau eine KKNF.

Beispiel aus der Tabelle (6 Folien vorher):

$$y_a = (x_2 \vee x_1 \vee x_0) \wedge (x_2 \vee x_1 \vee \overline{x_0}) \wedge (\overline{x_2} \vee x_1 \vee x_0) \wedge (\overline{x_2} \vee x_1 \vee \overline{x_0}) \wedge (\overline{x_2} \vee \overline{x_1} \vee x_0) \wedge (\overline{x_2} \vee \overline{x_1} \vee \overline{x_0})$$
$$= M_0 \wedge M_1 \wedge M_4 \wedge M_5 \wedge M_6 \wedge M_7$$
$$y_b = (x_2 \vee x_1 \vee \overline{x_0}) \wedge (\overline{x_2} \vee \overline{x_1} \vee \overline{x_0}) = M_1 \wedge M_7$$

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Konjunktive Normalform – DNF
(Allgemeine) konjunktive Normalform - KKNF

- Eine **konjunktive Normalform (KNF)** einer Schaltfunktion ist die **Konjunktion (UND-Verknüpfung) von ODER-Verknüpfungen von Eingangsvariablen** (in normaler oder negierter Form).
- Eine KNF enthält:
  - zwischen 1 und 2^n ODER-Verknüpfungen von jeweils 2 bis n Eingangsvariablen
  - sowie i. a. eine UND-Verknüpfung der Ergebnisse der ODER-Verknüpfungen.
- Zu einer Schaltfunktion gibt es neben der KKNF meist weitere Darstellungen in KNF.
- Eine KNF, die eine Schaltfunktion mit **minimalem Gatteraufwand** realisiert, heißt **konjunktive Minimalform (KMF).**

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Konjunktive Normalformen

Übungsaufgabe:

Schaltnetz, das die Funktion y_b in KKNF realisiert.

Alle Eingangsvariablen werden der Übersichtlichkeit halber negiert und dann von den 6 Signalen die benötigten an die ODER-Gatter der 1. Stufe gelegt. „Bus-artige“ Darstellung vorteilhaft.

Die Ausgänge aller ODER-Gatter werden an die Eingänge des einen UND-Gatters der 2. Stufe gelegt.

→ Siehe Tafel

Weitere:

- Funktionen y_a, y_c in KKNF.
- Gegeben ist die WT einer Schaltfunktion.
  Ist die Realisierung als KDNF oder KKNF günstiger?

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Normalformen – allgemeine Betrachtungen

- Die **kanonischen Normalformen** sind sehr einfach aus der **Wahrheitstabelle (WT)** einer Schaltfunktion abzuleiten.
- Der **Aufwand** zur Realisierung einer Schaltfunktion durch eine Gatteranordnung entsprechend einer kanonischen Form ist in der Regel **nicht minimal**.
- Die **disjunktiven Normalformen** kommen den **menschlichen Denkstrukturen** entgegen.
- Sofern alle Eingangsvariablen in normaler und negierter Form verfügbar sind, führt die KDNF genau dann auf eine einfachere Realisierung einer Schaltfunktion, wenn die Schaltfunktion für weniger als die Hälfte der möglichen Werte des Eingangsvektors eine 1 liefert, sonst die KKNF.

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen - Ziel

Ziel: Darstellung der Schaltfunktion in einer Form, die zu minimalem Aufwand (Kosten) bei der technischen Realisierung (meist in Form einer integrierten elektronischen Schaltung, IC) führt.

- Ausgegangen wird von einer **Schaltfunktion**, die **in** einer **beliebigen Beschreibungsform** gegeben ist.
- Unterschiedliche **Minimierungsziele** üblich, z. B.
  - minimale Gatterzahl
  - minimale Anzahl von Eingängen der Gatter
  - Abbildung auf vorhandene Gatterstrukturen,
    z. B. UND/ODER/NICHT, NANDs, NORs, Elemente in einer bestimmten Standardzell-Bibliothek, …
  - minimale Kosten
  - minimale Verlustleistung
  - maximale Geschwindigkeit
  - …

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen

- **Einfachstes Aufwandsmaß:**
  - Anzahl der notwendigen Verknüpfungsglieder (Gatter) und
  - die Gesamtzahl der erforderlichen Eingänge aller Gatter.
- **CSIC (customer-specific integrated circuit) und ASIC (application-specific integrated circuit):**
  → Minimierungsaufgabe sehr komplex (Flächenbedarf Chip etc.)
- **Progr. IC wie PALs, GALs, CPLDs oder FPGA:**
  → Minimierung mit Blick auf Grundgatter

- **Im folgenden**: Minimierung der Zahl der Gatter und ihrer Eingänge. (Bildung des Komplements nicht im Fokus)
- **Ergebnisse** sind hier stets zweistufige Gatteranordnungen:
  - als **disjunktive Normalform** (DNF) oder
  - als **konjunktive Normalform** (KNF)

  der Schaltfunktion realisieren.

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen
1. Anwendung der Schaltalgebra

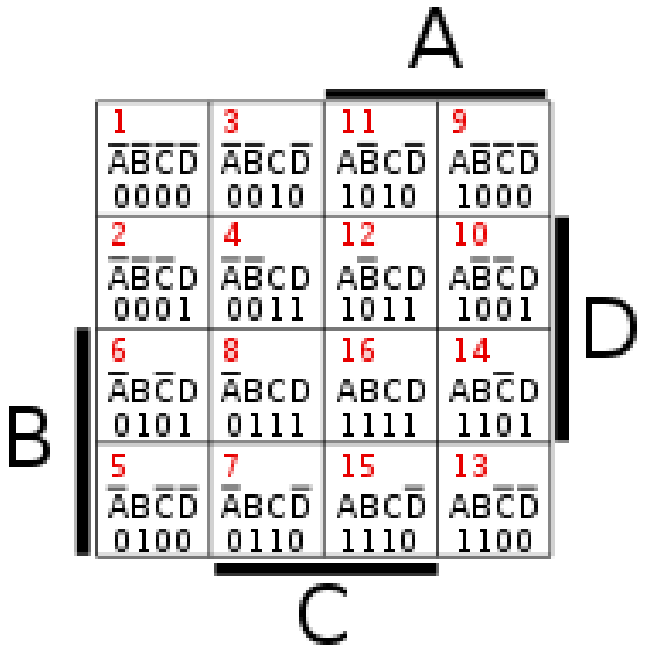
- Nutzen der Gesetze zum Zusammenfassen (**wie** in der **Mathematik**)
- In der **Praxis** völlig **irrelevant**.
- … lassen wir **weg; haben sie rudimentär im Tutorium gemacht**

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen
2. Grafisches Verfahren – KV-Diagramme

- **Grafisches Verfahren** = sehr **anschaulich** und „**selbsterklärend**“.
- In der **Praxis** nahezu **irrelevant**.
- Machen wir **trotzdem**, um die nicht grafischen Verfahren zu verstehen.
- Im folgenden **max. 4-Bit-Eingangsvariabel**
- Ergebnis ist immer eine disjunktive oder konjunktive Normalform:
  - disjunktive **Minimalform** (DMF) oder
  - konjunktive **Minimalform** (KMF).



[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen, Wikipedia]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen
2. Grafisches Verfahren – KV-Diagramme

- **Zusammenfassen** von **logisch benachbarten Feldern** (HD = 1) zu **Blöcken** mit **Kantenlängen, die 2er-Potenzen sind: 1, 2, 4**
- **Blöcke** dürfen sich **überlappen.**
- Bei **2er/4er/8er-Feldern entfallen 1/2/3 Variable** pro ehemaligem **Minterm/Maxterm**.
  Aus einer kanonischen Form wird eine Minimalform.
- Evtl. gibt es **mehrere verschiedene Minimalformen** eines Typs. Wenn man die Wahl hat, sollte man wegen des stabileren dynamischen Verhaltens überlappende statt nicht überlappende Blöcke wählen.
- **„Don‘t care“-Einträge erleichtern** meist die **Minimierung**.

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen
2. Grafisches Verfahren – KV-Diagramme

Disjunktive Minimalform (DMF):

- Zusammenfassung aller Felder mit „1“ zu Blöcken
- Beschreibung jedes einzelnen Blocks durch UND-Verknüpfung der notwendigen Eingangsvariablen (normal oder invertiert)
- Schaltfunktion = ODER-Verknüpfung der einzelnen UND-Verknüpfungen.

- Felder mit don’t care-Symbolen („X“) dürfen zu Blöcken als „1“-Feld verwendet werden, falls sinnvoll - ansonsten als „0“.

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen
2. Grafisches Verfahren – KV-Diagramme

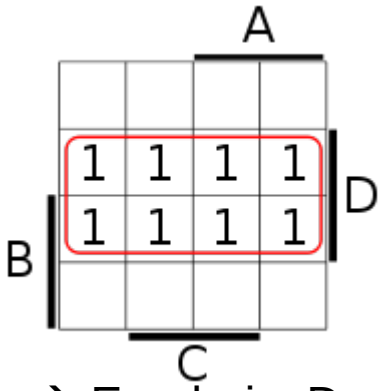
Disjunktive Minimalform (DMF) - Regeln:

- Benachbarte Felder mit „1“ = Blöcke
- Alle „1“ **müssen** in Gruppen zusammengefasst werden.
- Benachbarte Felder mit Einsen werden zu Blöcke zusammengefasst.
- Blöcke müssen so groß wie möglich sein.
- So wenig Blöcke wie möglich
- Die Blöcke dürfen nur Größen haben, die Zweierpotenzen entsprechen.
- Die Blöcke müssen rechteckige Blöcke sein.
- Die Blöcke dürfen sich überlappen.
- Die Blöcke dürfen über die Ränder hinweggehen.
- Zwei Blöcke dürfen nicht exakt die gleichen Einsen umfassen.
- Es darf keine Gruppe vollständig von einer anderen Gruppe umschlossen werden.

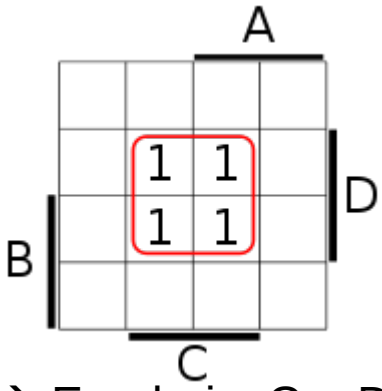
[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

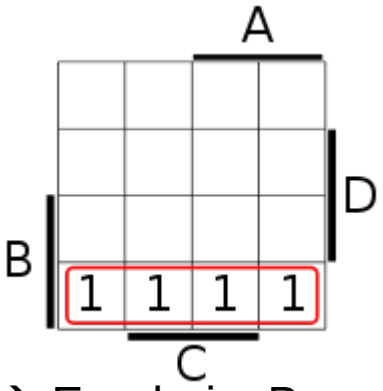
Minimierung von Schaltfunktionen
2. Grafisches Verfahren – KV-Diagramme
Disjunktive Minimalform (DMF) – Beispiele mit Eingangsvariablen ABCD:



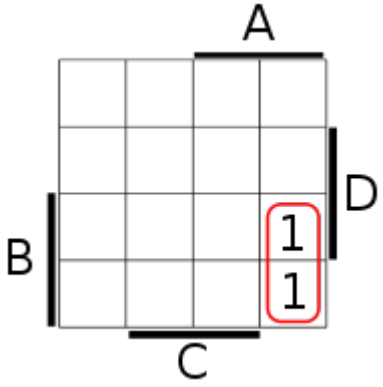
→ Ergebnis: D



→ Ergebnis: C ∧ D

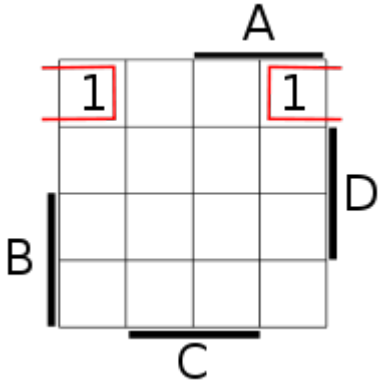


→ Ergebnis: B ∧ ¬D



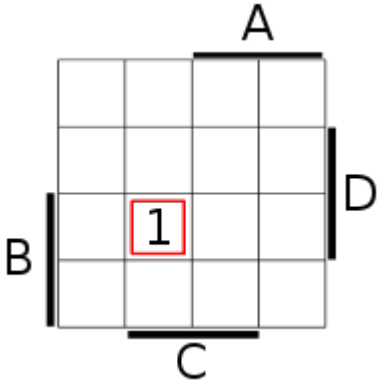
→ Ergebnis:

$A \wedge B \wedge \neg C$



→ Ergebnis:

$\neg B \wedge \neg C \wedge \neg D$



→ Ergebnis:

$\neg A \wedge B \wedge C \wedge D$

[Quelle: Wikipedia]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen
2. Grafisches Verfahren – KV-Diagramme

Konjunktive Minimalform (DMF):

- **Zusammenfassung** aller Felder mit **„0“ zu Blöcken**
- Dann **zwei Varianten**:
  1. → Jeder einzelne Block wird wie bei DMF durch eine UND-Verknüpfung der Eingangsvariablen beschrieben.
     → Komplementierte Schaltfunktion ergibt sich aus der ODER-Verknüpfung der einzelnen UND-Verknüpfungen.
     → Für die Schaltfunktion selbst = Komplementierung beider Seiten
  2. → Jeder einzelne Block wird durch eine ODER-Verknüpfung der komplementierten zugehörigen Eingangsvariablen beschrieben.
     → Schaltfunktion = UND-Verknüpfung der einzelnen ODER-Verknüpfungen.
- Felder mit don’t care-Symbolen („X“) dürfen zu Blöcken als „0“-Feld verwendet werden, falls sinnvoll - ansonsten als „1“.

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen
2. Grafisches Verfahren – KV-Diagramme

Übungsaufgabe:
Die beiden durch KV-Diagramme beschriebenen Schaltfunktionen sind jeweils in eine DMF und in eine KMF umzuwandeln:

a)

x_2 \ x_0				
	1	1	0	1
	1	0	1	1
	1	0	1	1
	1	1	0	1

x_3 / x_1

b)

x_2 \ x_0				
	1	0	0	0
	X	1	1	X
	X	0	X	0
	1	X	0	1

x_3 / x_1

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]

Prof. Dr.-Ing. Thorsten Uelzen
Fakultät Elektrotechnik - Wolfenbüttel
T.Uelzen@Ostfalia.de

Minimierung von Schaltfunktionen
3. Nichtgrafische Verfahren

- **Sehr viele verschiedene Programme** und Algorithmen
- In der Praxis die **häufigste Methode**; oft Teil der Logik-Synthese.
- Anwendbar auf **beliebig viele Eingangsvariablen (hunderte und tausende)**
- **Wichtige Vertreter**:
  - Quine-McCluskey-Algorithmus (W. V. Quine und E. J. McCluskey)
  - ESPRESSO-Algorithmus (R. Brayton / IBM entwickelt, 1984)
- **Extrem komplexes mathematisches Problem = extremer Rechenaufwand**
- In der Praxis oft **weitere Randbedingungen** (constraints) wichtig:
  - Maximale Signallaufzeiten für einzelne Signale
  - Signallaufzeitunterschiede (Hazards, Glitches, Spikes)
  - Begrenzung des Leistungsbedarfes
  - Testbarkeit der Schaltung nach der Herstellung (und ggf. im Betrieb)
  - Verwendung von „Vorzugs“-Gattern (z. B. Standardzellen)

[Quelle: Harriehausen, T: Arbeitsblätter zu Digitaltechnik Grundlagen]