

Technische Informatik



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi
FACHBEREICH INFORMATIK

Digitaltechnik II

Prof. Dr. Ralf S. Mayer,
Fachbereich Informatik, h_da
Sommersemester 2007

Organisatorisches

- Prof. Ralf Mayer, Raum 1.07, Gebäude D14
- Web: <http://www.fbi.h-da.de/~rmayer>
- Email: r.mayer@fbi.h-da.de
- Sprechzeiten
nach Vereinbarung
- **Praktikum**: Start 04.04.2007 Gruppe y
- **Klausur** am 09.07.2007 08:30-10:00
B11/3/4
 - Online Belegung der Klausur bis **24.06.2007**
 - Abmeldung bis 05.07.2007



0.1 Vorlesung Digitaltechnik

- 2-semesterige Veranstaltung
- Digitaltechnik 1: Vorlesung,
 - Leistungsnachweis Klausur (**Schein**)
- Digitaltechnik 2: Vorlesung und begleitendes Praktikum,
 - erfolgreiches Praktikum ist Voraussetzung für Klausurzulassung,
 - Leistungsnachweis Klausur (**Schein**)

0.2 Praktikum Digitaltechnik II

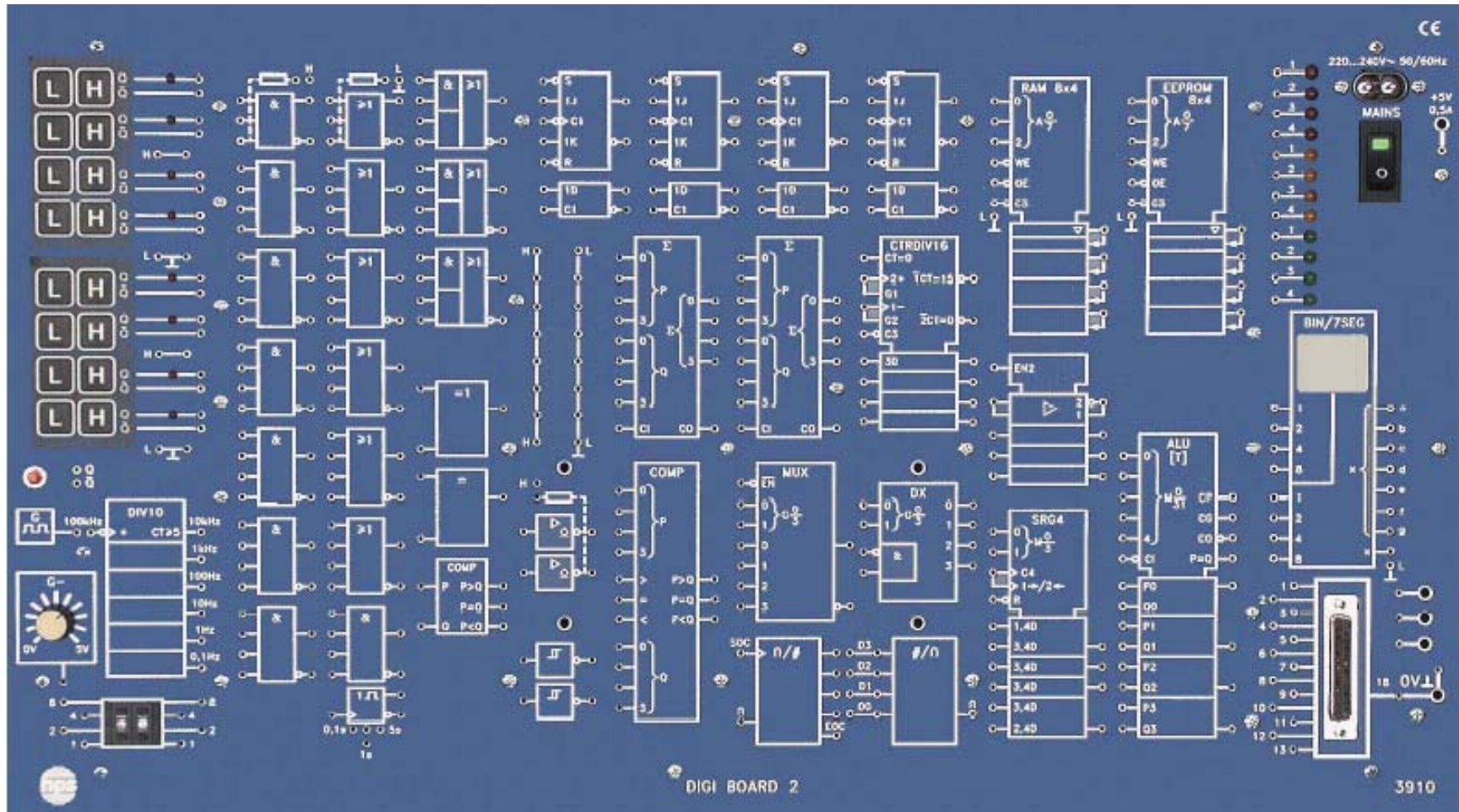
- Betreuung: Michael Müller, Ralf Mayer
- Ort: D10 / 0.31
- Uhrzeiten: s. OBS
- Ablauf:
 - 2er Gruppen
 - Vorbereitung jedes Versuchs
 - Schaltungsaufbau erklären
 - Schaltung funktioniert → Abnahme (Testat)
 - Nach erfolgreicher Durchführung, kurze Ausarbeitung → Blatt
 - **neu**: Regeln (Nichteinhaltung → Minuspunkte ... → *Ausschluss*)
- 5 Pflichtaufgaben (Termine)
 - Aufgaben etc.:
<http://www.fbi.h-da.de/organisation/personen/kasper-klaus/digitaltechnik-2/materialien-zur-veranstaltung-digitaltechnik-2.html>
 - Anwesenheitspflicht (Krankheit → Attest)

0.2 Praktikum Digitaltechnik II (2)

- Anzahl Teilnehmer (aktuell):
- 16 Do5x D10/031
- 16 Do5y D10/031
- 15 Mi4x D10/031
- 4 Mi4y D10/031

Wer möchte nach Mi4y tauschen?

0.2 Praktikum Digitaltechnik II (2)



Digi-Board: Anleitung siehe Web-Links

- Skript Prof. J. Wietzke (nur pers. Gebrauch)
http://www.fbi.h-da.de/fileadmin/personal/j.wietzke/mein_ordner/DT/Skripte/
user: pwd:
- Informationen, Folien zur Vorlesung, etc.
<http://www2.fbi.h-da.de/~rmayer/DT2>
Bei geschützten Dokumenten:
User:
Pwd:
- Simulation digitaler Schaltungen:
LOGIX (HS-Lizenz, per USB-Stick bei Herrn Müller)
<http://www.digitalsimulator.de> (? Schaltskizzen ??!)
Linux : <http://ksimus.berlios.de/> (*n.n. getestet*)
- Literatur

0.4 Literatur (alphabetisch)

■ **Computerarchitektur**

Andrew S. Tanenbaum, James Goodman 29,95 €
ISBN 3-8273-7016-7, Verlag Pearson Studium

■ **Digitaltechnik**

K. Beuth, O. Beuth 34,80 €
ISBN 3-8023-1958-3 12. Auflage 2003 Vogel Buchverlag 2003

■ **Technische Informatik 1**

Schiffmann – Schmitz 29,95 €
ISBN 3-540-40418-X 5. Auflage Springer-Verlag Berlin Heidelberg
New York

■ **Technische Informatik 2**

Schiffmann 34,95 €
ISBN 3-540-22271-5 5. Auflage Springer-Verlag Berlin Heidelberg
New York

■ **Weitere Literatur s. meine Webseite**

■ **Skript !**

[http://www.fbi.h-da.de/fileadmin/personal/
j.wietzke/mein_ordner/DT/Skripte/DT2.pdf](http://www.fbi.h-da.de/fileadmin/personal/j.wietzke/mein_ordner/DT/Skripte/DT2.pdf)

0.6 Struktur der Vorlesung

- Vorlesung
 - Wiederholung
 - neue Themen
- Nachbearbeitung
 - Skripte, Literatur
 - Aufgaben, Übungen, Simulations-Software, ...
- Fragen im Rahmen der Wiederholung
- Klausurvorbereitung

0.7 Organisation Ihres Studiums

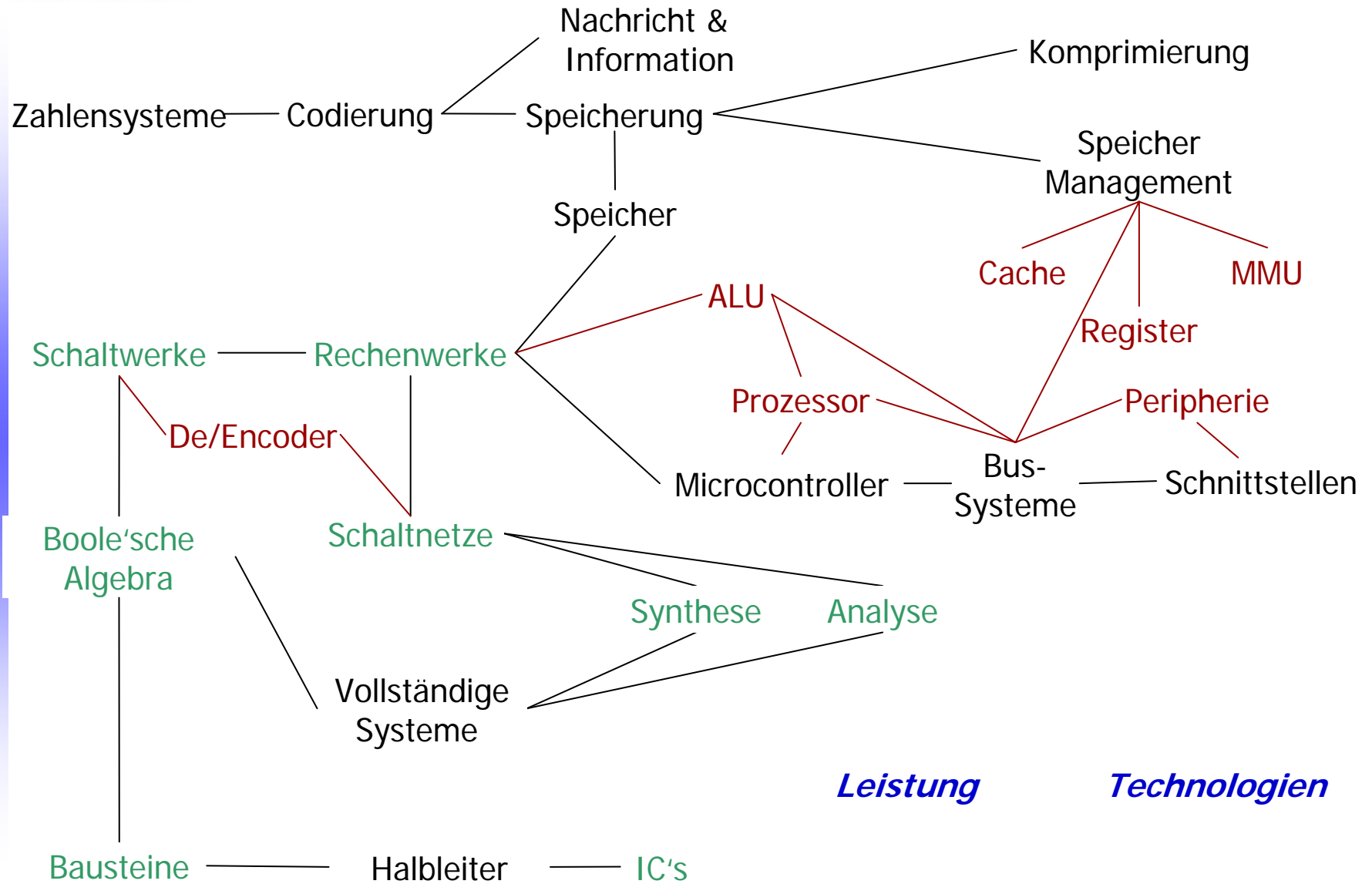
- Selbstständigkeit
- Eigenverantwortlichkeit
- Erkenntnis
 - Thematik verstanden
 - gestellte Aufgaben lösen
 - sinnvolle Aufgabenstellungen generieren
- Lerngruppen

- Mentorensystem (neu am Fachb. Informatik)

- Boolesche Algebra
 - Grundlagen
 - Rechenregeln
 - KV-Diagramme
- Schaltalgebra (Gatter z.B. AND, OR, NOR)
- Bauelemente / Elektronik
 - Halbleiter
 - Logic-Familien (TTL, CMOS, ...)
- Schaltnetze
 - Analyse / Synthese
 - Addierer, Multiplizierer

9. Einleitung
10. Sequentielle Grundschaltungen, Schaltwerk
(Flip-Flop, Schieberegister)
13. Programmierbare Bausteine
(EPROM, FPGA)
14. Speicherbausteine
(RAM, ROM)
17. Codierverfahren
(Huffman-Codierung, Hamming-Distanz)
18. Massenspeicher
(HDD, CD, DVD)
19. Automaten

Lerninhalte DT2, RG und MPS



0. Organisatorisches
9. Einleitung
10. Sequentielle Grundschaltungen, Schaltwerke
11. Spezielle Schaltwerke
12. Hazards
13. Programmierbare Bausteine
14. Speicherbausteine als programmierbare Logik
15. FPGA
16. Speicher
17. Codierverfahren und Codesicherung
18. Massenspeicher
19. Automaten

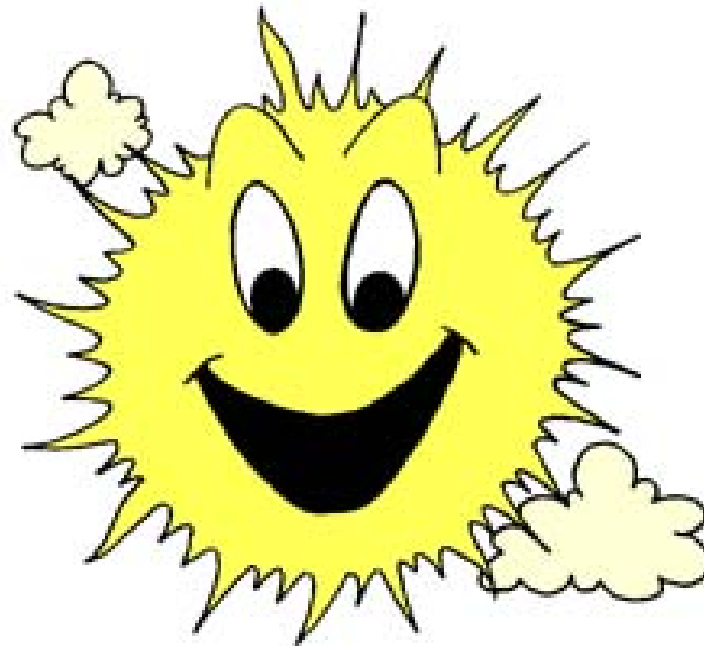
0.9 Einleitung

Bei dieser Vorlesung wird meist in der Reihenfolge des Skripts vorgegangen.

Abweichungen davon, wie auch Vertiefung oder Kürzung einzelner Inhalte sowie Zusammenfassung in anderen Kapiteln sind möglich.

Sofern nicht explizit angegeben, ist das Skript inhaltlich relevant für den Leistungsnachweis.

Wiederholung aus Digitaltechnik I



2.3-2.5. Übersicht Boolesche Gleichungen

Gesetz oder Postulat	UND-Form	ODER-Form
P1	A=0 oder A=1 ; $\{A,B,C,\dots\} \in$ Boolesche Menge	
P2	$0 \wedge 0 = 0$	
P3	$1 \wedge 1 = 1$	
P4		$0 \vee 0 = 0$
P5		$1 \vee 1 = 1$
P6	$1 \wedge 0 = 0 \wedge 1 = 0$	
P7		$1 \vee 0 = 0 \vee 1 = 1$
P8	$\neg 1 = 0$ und $\neg 0 = 1$	
NULL-Gesetz	$A \wedge 0 = 0$	$A \vee 0 = A$ (Identitätsgesetz)
EINS-Gesetz	$A \wedge 1 = A$ (Identitätsgesetz)	$A \vee 1 = 1$
Idempotenzgesetz	$A \wedge A = A$	$A \vee A = A$
Inversionsgesetz	$A \wedge \neg A = 0$	$A \vee \neg A = 1$
Kommutativgesetz	$A \wedge B = B \wedge A$	$A \vee B = B \vee A$
Assoziativgesetz	$A \wedge (B \wedge C) = (A \wedge B) \wedge C$	$A \vee (B \vee C) = (A \vee B) \vee C$
Distributivgesetz	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
Absorptionsgesetz	$A \wedge (A \vee B) = A$	$A \vee (A \wedge B) = A$
De Morgansche Gl.	$\neg(A \wedge B) = \neg A \vee \neg B$	$\neg(A \vee B) = \neg A \wedge \neg B$

2.4.7 Kürzungsregeln

1. $X_1 \vee (X_1 \wedge X_2) = X_1$
2. $X_1 \wedge (X_1 \vee X_2) = X_1$
3. $X_1 \vee (\overline{X_1} \wedge X_2) = X_1 \vee X_2$
4. $X_1 \wedge (\overline{X_1} \vee X_2) = X_1 \wedge X_2$
5. $(X_1 \wedge X_2) \vee (X_1 \wedge \overline{X_2}) = X_1$
6. $(X_1 \vee X_2) \wedge (X_1 \vee \overline{X_2}) = X_1$

2.5 De Morgan'sche Gesetze

$$1. \quad \overline{(X_1 \wedge X_2)} = (\overline{X_1} \vee \overline{X_2})$$

$$2. \quad \overline{(X_1 \vee X_2)} = (\overline{X_1} \wedge \overline{X_2})$$

2.6 Dualitätsprinzip, Shannon'sche Gesetz

$$\overline{f(X_1, X_2, \dots, X_n, \wedge, \vee)} = f(\overline{X_1}, \overline{X_2}, \dots, \overline{X_n}, \vee, \wedge)$$

Der **invertierte Wert einer Booleschen Funktion** ist gleich dem Wert, die diese Funktion liefert, wenn alle **Operanden** und alle **Operatoren invertiert** werden.

5.1 Normalformen

Merke:

- **Disjunktion** = **ODER** – Verknüpfung
- **Konjunktion** = **UND** – Verknüpfung

5.1 Normalformen

- Normalformen dienen dazu, eine beliebige Funktion in einheitlicher Form zu beschreiben.
- Normalformen beschreiben eine Schaltfunktion ausgehend von einer Wertetabelle in Gleichungsform
- In der booleschen Algebra sind zwei Normalformen gebräuchlich
 - Disjunktive Normalform (DNF)
 - Konjunktive Normalform (KNF)
- Normalformen basieren auf
 - Mintermen (m_i)
 - Maxtermen (M_i)

5.2 Minterme (Vollkonjunktion)

- **Minterme** sind **UND**-Verknüpfungen, die **alle** Schaltvariablen einmal enthalten
- Jede Variable kann sowohl **negiert**, als auch **nicht negiert** vorkommen

Wenn x_1, x_2, \dots, x_n die gegebenen Variablen sind,
und \tilde{x}_i sowohl x_i als auch $\overline{x_i}$, dann ist

$$\boxed{\tilde{x}_1 \wedge \tilde{x}_2 \wedge \dots \wedge \tilde{x}_n} \text{ ein } \mathbf{Minterm}$$

Bei n Schaltvariablen gibt es
 2^n verschiedene Minterme m_i

5.2 Minterme n=2

x_2	x_1	$x_2 \wedge x_1$	$x_2 \wedge \bar{x}_1$	$\bar{x}_2 \wedge x_1$	$\bar{x}_2 \wedge \bar{x}_1$
0	0	0	0	0	①
0	1	0	0	①	0
1	0	0	①	0	0
1	1	①	0	0	0

Kombination aller Minterme aus n=2

Bei den einzelnen Kombinationen werden **Variablen mit dem Wert 0** negiert, um für den jeweiligen **Minterm m_i** den Wert **1** zu erzeugen.

5.4 Disjunktive Normalform (DNF)

- Eine beliebige boolesche Funktion lässt sich durch die **disjunktive Verknüpfung** geeigneter **Minterme** realisieren.
- Bei vorgegebener Funktion nimmt man genau die **Minterme**, welche den Wert '1' an den Stellen erzeugen, an denen die Funktion '1'-Werte ausgibt. **Variablen**, die **bei dieser Kombination** den Wert **0** haben **werden negiert**.
- Durch die **disjunktive Verknüpfung** dieser **Minterme** erhält man **genau die vorgegebene Funktion** → **Disjunktive Normalform (DNF)**

5.3 Maxterme (Volldisjunktion)

- **Maxterme** sind **ODER**-Verknüpfungen, die **alle** Schaltvariablen einmal enthalten
- Jede Variable kann sowohl **negiert**, als auch **nicht negiert** vorkommen

Wenn x_1, x_2, \dots, x_n die gegebenen Variablen sind,
und \tilde{x}_i sowohl x_i als auch $\overline{x_i}$, dann ist

$\tilde{x}_1 \vee \tilde{x}_2 \vee \dots \vee \tilde{x}_n$ ein **Maxterm**

Bei n Schaltvariablen gibt es
 2^n verschiedene Maxterme M_i

5.3 Maxterme n=2

x_2	x_1	$x_2 \vee x_1$	$x_2 \vee \bar{x}_1$	$\bar{x}_2 \vee x_1$	$\bar{x}_2 \vee \bar{x}_1$
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

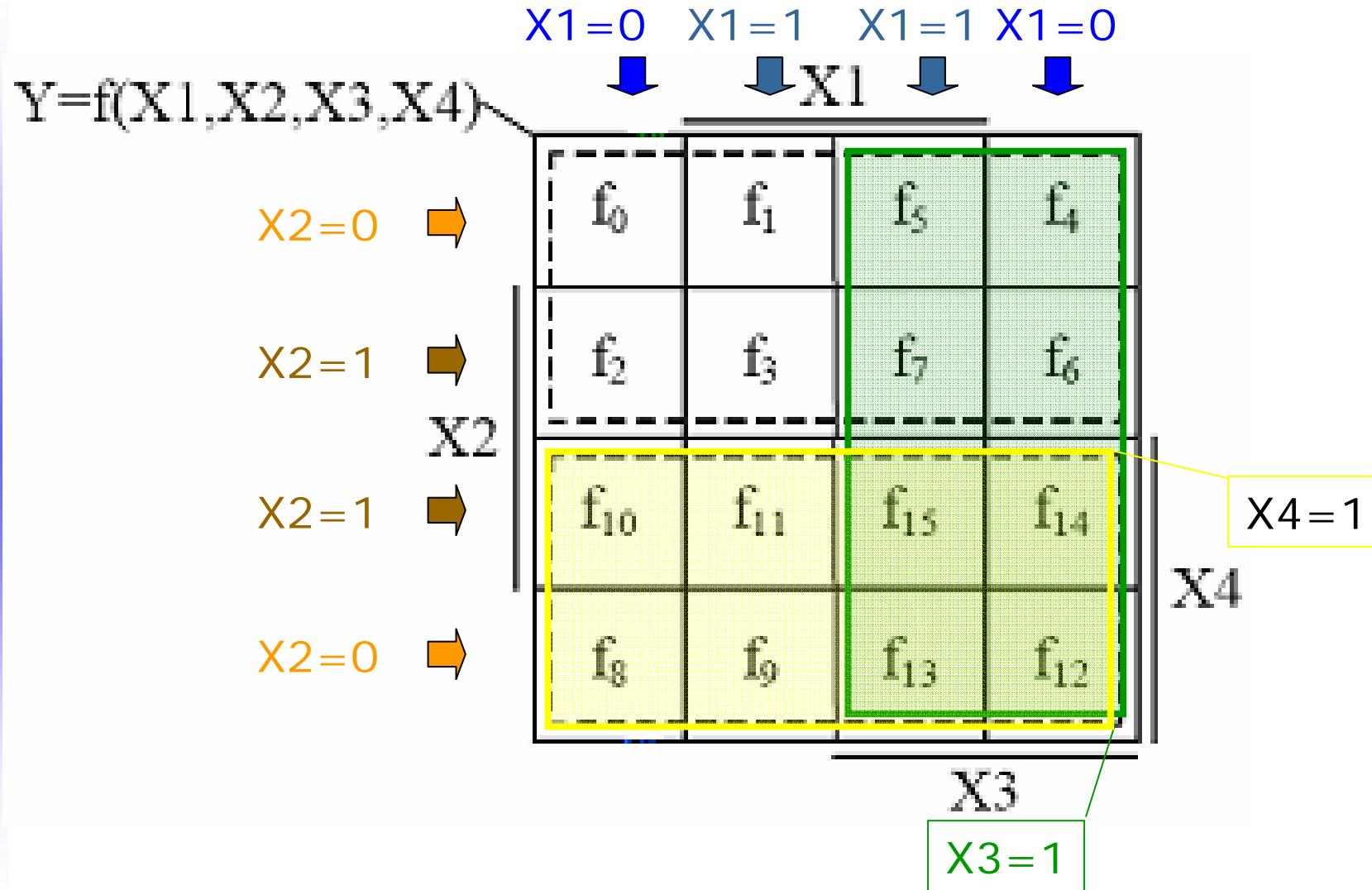
Kombination aller Maxterme aus n=2

Bei den einzelnen Kombinationen werden **Variablen mit dem Wert 1 negiert**, um für den jeweiligen **Maxterm M_i** den Wert **0** zu erzeugen.

5.5 Konjunktive Normalform (KNF)

- Eine beliebige boolesche Funktion lässt sich durch **konjunktive Verknüpfung** geeigneter **Maxterme** beschreiben.
- Für jede Eingangskombination, an der die Funktion den Ausgangswert '0' produziert, verknüpft man die zugehörigen **Maxterme** mittels Konjunktion. **Variablen**, die **bei dieser Kombination** den Wert **1** haben **werden negiert**
- Durch die **konjunktive Verknüpfung** dieser **Maxterme** erhält man **genau die vorgegebene Funktion** → **Konjunktive Normalform (KNF)**

5.7 KV-Diagramm für n=4



5.7 Implikanten

Definition Implikant:

Fasst man Min- bzw. Maxterme einer Funktion so zusammen, dass deren Verbund durch einen Term geringerer Komplexität, d.h. mit einer reduzierten Anzahl von Eingangsvariablen, beschrieben werden kann, so wird der resultierende Term als **Implikant** bezeichnet.

Die **Anzahl** der in einem Implikanten **zusammengefassten Min- bzw. Maxterme** bildet eine **2er-Potenz**.

5.7 Primimplikanten (PI)

Ist ein Implikant einer booleschen Funktion in keinem anderen Implikanten vollständig enthalten, wird er als **Primimplikant** bezeichnet.

Bezeichnung für die so groß wie möglich gewählten Blöcke von „Einsen“ im Karnaugh-Diagramm (in der DNF)

5.7 Kernprimimplikanten (essentieller PI)

- Enthält ein Primimplikant (PI) mindestens einen **Min- bzw. Maxterm**, der **in keinem anderen Primimplikanten enthalten** ist, bezeichnet man diesen als **Kern-Primimplikanten**.
- → die **minimale Lösung** enthält **mindestens** die **essentiellen Primimplikanten** .
- **Nicht-essentielle PI** sind Primimplikanten, die keine Kern-Primimplikanten sind.
- **Redundante PI** sind nicht-essentielle PI die nur von essentielle PI bereits überdeckte „1“en bzw. „0“en enthalten .

5.7 Lokalisierung Implikanten im KV-Diagr.

	X1			
	0	1	1	0
X2	1	1	1	0
	X3			

$$I1 = (\neg X3) \wedge X2$$

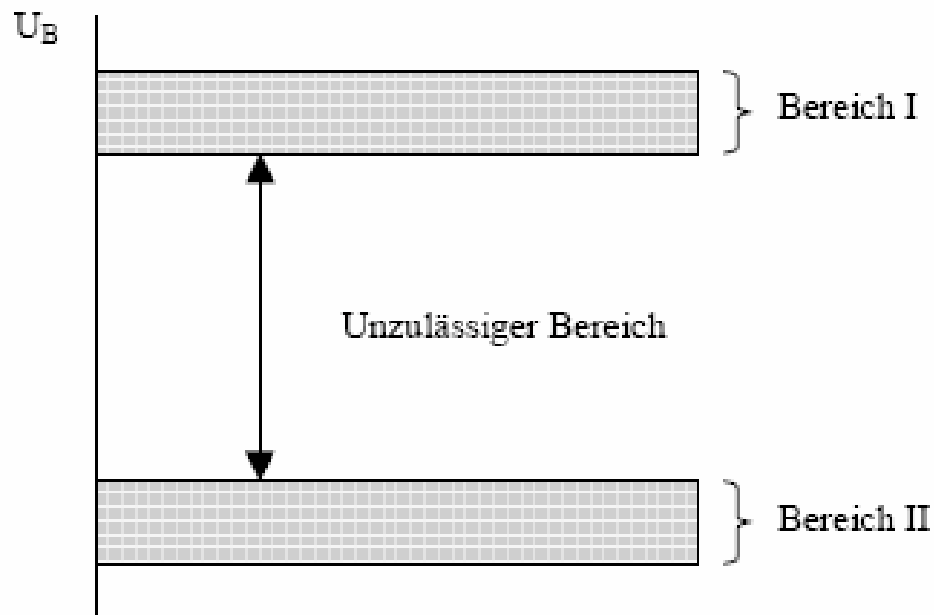
$$I2 = (\neg X3) \wedge X1$$

$$I3 = X1$$

- **Implikanten** bilden sich aus 2^i benachbarten Feldern
- Lässt sich ein Implikant nicht mehr mit benachbarten Implikanten zusammenfassen, ist dieser ein **Primimplikant**
- **I1** und **I3** sind **Kernprimimplikanten**, weil
 - Minterm m2 nur in I1
 - Minterme m1, m5 und m7 nur in I3
 enthalten sind.

7.0 Logik-Pegel

Bei der praktischen Umsetzung digitaler Schaltungen werden, aus Toleranzgründen (Temperatur, Versorgungsspannung, Last, etc.), Pegelbereiche für die zwei gültigen Zustände „wahr“ und „falsch“ angegeben:



Typische Versorgungsspannungen von IC's sind:

- TTL, 5V
 - L: $< 0.8 \text{ V}$
 - H: $> 2 \text{ V}$
- CMOS, 5-20V
- BICMOS, 5V/3,3V
- NMOS, 5V
- ECL, -5,2V.

Ende der Wiederholung

