Digitaltechnik

3. Schaltalgebra

Prof. Dr. Eckhard Kruse

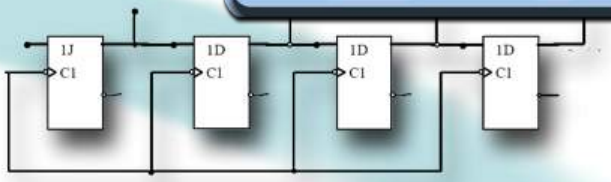
DHBW Mannheim

Vorlesungsthemen (s. Studienplan)

Elektronische Realisierung
- Elektronikgrundlagen
- Elementare Gatter
- Technologien (TTL, CMOS)
- ...

Standardbaugruppen
- Flip-Flops
- Zähler
- Schieberegister
- ...

Schaltalgebra
- Logische Verknüpfungen
- Gatter + Schaltnetze
- Schaltungstransformation

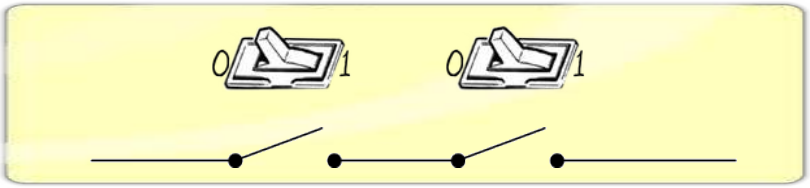


Übungen

Zahlentheorie
- Binärcodierung
- Hexadezimal usw.
- Binäres Rechnen

00100110 11101101
11011010 11101101
11101101 01110110
11110110 01110110
01110110 00100110
11011010 01110110

Konzepte und technische Umsetzung

Mathematik/Konzepte	Technische Umsetzung
Fließkommadarstellung, Zeichencodierung usw.	
Binärarithmetik 1011 x 101 1011 1011 110111	
Boolesche Algebra $1 \wedge 0 = 0$	
Binärsystem 0 1	„Strom an/aus“ 

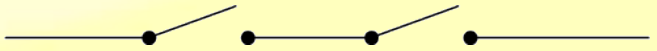
Basisoperationen

Mathematik/Konzepte

Technische Umsetzung

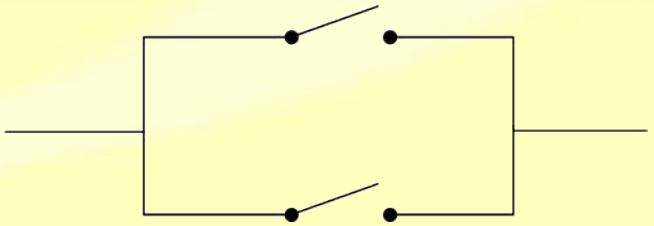
Und-Operation

$\wedge$	0	1
0	0	0
1	0	1



Oder-Operation

$\vee$	0	1
0	0	1
1	1	1



Schaltalgebra

Definition

Schaltalgebra ist die technische Anwendung der **Booleschen Algebra**. Beide basieren auf:

- Binären Werten:
  - 1 (bzw. wahr, true, high, Strom fließt, Spannung an …)
  - 0 (bzw. falsch, false, low, kein Strom, keine Spannung …)
- Operationen zur Verknüpfung
  - und, oder, nicht (bzw. and, or, not)

- Boolesche Algebra
  - Begründet von George Boole (1815 – 1864, engl. Mathematiker)
  - Ursprung in der Logik = Zweig der Mathematik, der sich mit Aussagen, deren Wahrheitsgehalt sowie Folgerungs- und Beweismechanismen befasst
- Vielfältige Verwendung in der Informatik: Grundlage aller (integrierten) Schaltungen, Zahlendarstellungen und Berechnungen, Verknüpfung von logischen Bedingungen in der Programmierung ...

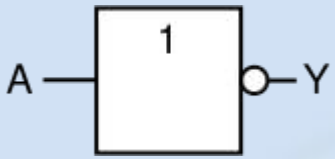
Grundoperationen

Definition

Nicht / Not

$Y = \neg A$

¬	0	1
	1	0



Und / And

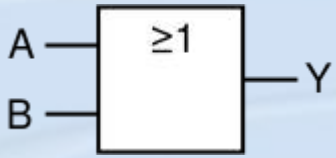
$Y = A \wedge B$

∧	0	1
0	0	0
1	0	1

Oder / Or

$Y = A \vee B$

∨	0	1
0	0	1
1	1	1

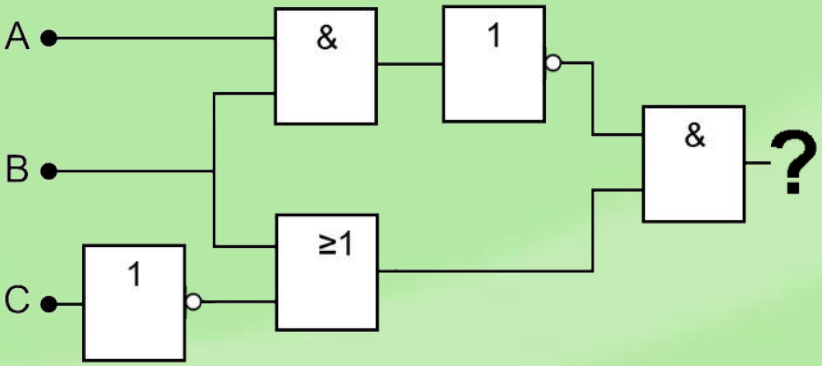


Übung

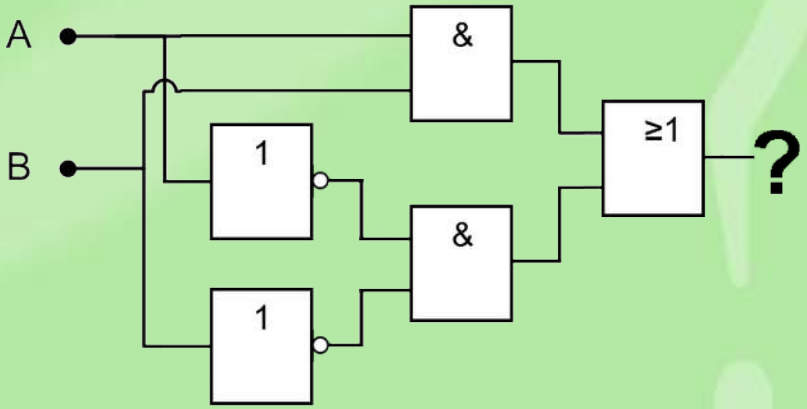
3.1 Schaltnetze

Geben Sie die Ausgabewerte der folgenden Schaltnetze an.

a) A=1, B=0, C=0



b) A=0, B=1

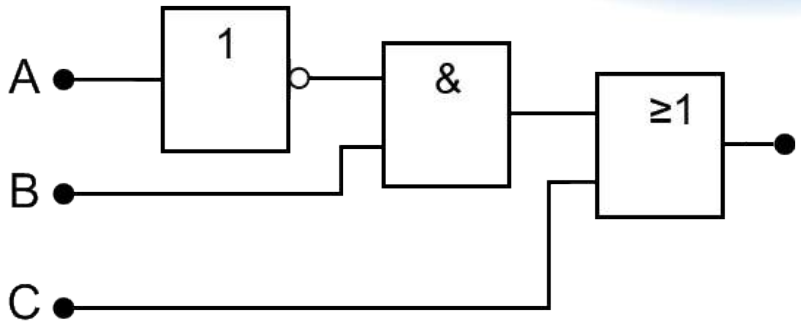


c) Variieren Sie die Eingabewerte. Wie verändern sich die Ausgabewerte?

Wahrheitstabellen

Definition

Wahrheitstabellen ordnen Kombinationen von Eingabewerten Ausgabewerte zu.



A	B	C	$\neg A$	$\neg A \wedge B$	$(\neg A \wedge B) \vee C$
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	0	0	1

Wahrheitstabelle

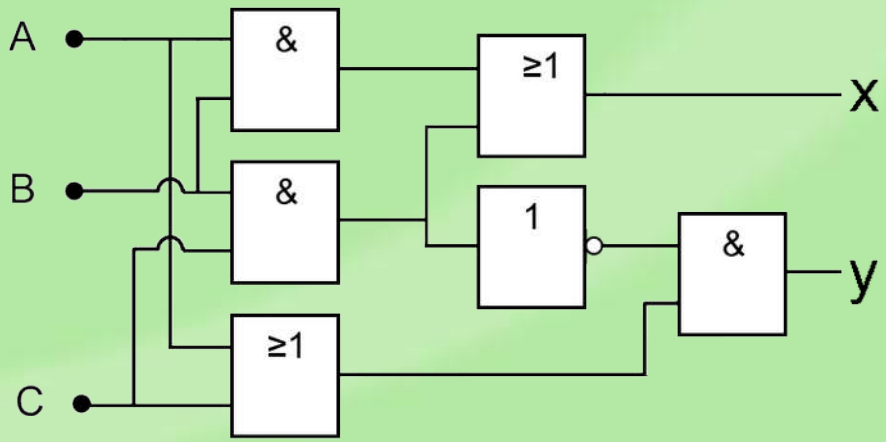
Übung

Übung

3.2 Wahrheitstabelle

Geben Sie die Wahrheitstabelle des folgenden Schaltnetzes an:

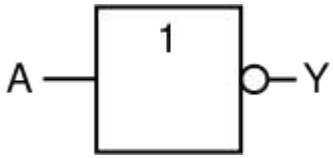
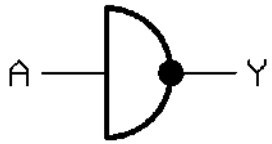
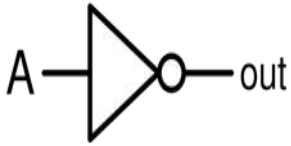
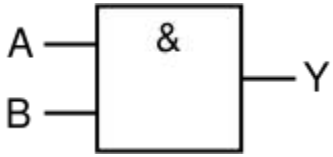
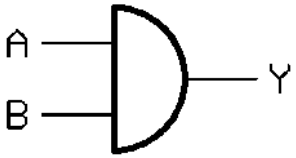
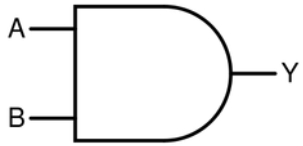
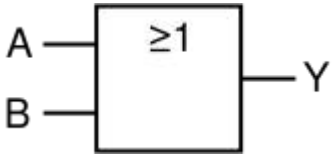
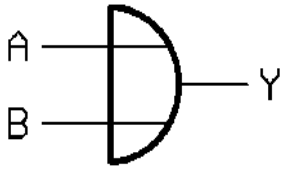
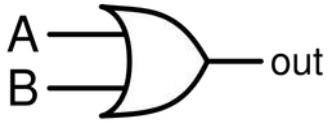
a)



b) Optional: Für die Schaltnetze der Aufgabe 3.1.

Nicht verwirren lassen!

Für die Gatter gibt es auch andere Symbole:

	DIN EN 60617 (40900)	DIN 40700 (alt)	US ANSI 91-1984
NOT			
AND			
OR			

Andere Schreibweisen der Booleschen Operatoren:
(je nach Kontext, z.B: Programmiersprache, Gattersymbole etc.):

$\neg$: ~, — $\wedge$: $\cap$ & • UND $\vee$: $\cup$ + ≥1 ODER

Gatter-Varianten

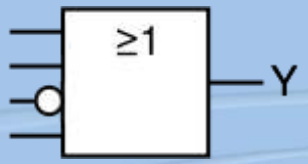
Verknüpfung von NOT mit AND, OR:

NAND

NOR

XOR: Exklusiv-Oder: (Schreibe: A $\veebar$ B)

Mehr als zwei Eingänge, negierte Eingänge



Wie lauten die Wahrheitstabellen dieser Gatter?

Übung

Übung

3.3 Simulationswerkzeug

Mit Hilfe geeigneter Software lassen sich Schaltnetze (und auch komplexere digitale Schaltungen) bequem simulieren. Hierzu soll ein Simulationswerkzeug installiert und ausprobiert werden.

a) Installieren Sie eine Digitalsimulator-Software, z.B.:
   `http://www.ksquared.de/jdigitalsimulator/`
   `https://sourceforge.net/projects/digisimulator/files/install%20EXE/5.57/`
   `http://www.logiccircuit.org/`
   s.a.:
   `http://www.mikrocontroller.net/articles/Schaltungssimulation`

b) Geben Sie einfache Schaltnetze ein (z.B. aus den vorigen Übungen) und experimentieren Sie mit dem Programm.

c) Welche Erfahrungen haben Sie mit dem Programm gemacht? Was finden Sie gut, was finden Sie nicht so gut?

Ampel

Übung

Übung

3.4 Ampel

Die Zustände einer Ampel lassen sich mit 2 Bit kodieren. Entwerfen Sie ein Schaltnetz, welches darauf basierend über drei Ausgänge die Lampen der Ampel ansteuert.

a) Definieren Sie die Ampelzustände und Bitkodierung.

b) Erstellen Sie eine Wahrheitstabelle für die Ampelschaltung.

c) Entwerfen (und simulieren) Sie ein geeignetes Schaltnetz.

d) Optional: Kodieren Sie die Zustände so, dass bei den Ampelzustandswechseln immer nur ein Bit geändert werden muss (→ Gray-Code). Entwerfen Sie das entsprechende Schaltnetz.

Systematisches Vorgehen?

- Wieviele Wertekombinationen (= Zeilen in der Wahrheitstabelle) gibt es für n Eingangswerte?
- Wieviele verschiedene Schaltfunktionen gibt es bei n Eingangswerten?
- Wie lassen sich Schaltungen für beliebige Wahrheitstabellen systematisch entwickeln?
- Genügt ein Gattertyp, um beliebige Schaltfunktionen zu realisieren?
- Wie lassen sich Schaltungen systematisch entwickeln und vereinfachen?

Spontane Antworten?

Kombinationen zweier boolscher Variablen

x_1	x_2	0	$x_1 \wedge x_2$	$x_1 \wedge \overline{x_2}$	x_1	$\overline{x_1} \wedge x_2$	x_2	$(x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2)$	$x_1 \vee x_2$
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
		Null	Konjunktion AND	Inhibition	Transfer	Inhibition	Transfer	Antivalenz XOR	Disjunktion OR

x_1	x_2	1	$\overline{x_1 \wedge x_2}$	$\overline{x_1} \vee x_2$	$\neg x_1$	$x_1 \vee \overline{x_2}$	$\neg x_2$	$(\overline{x_1} \vee x_2) \wedge (x_1 \vee \overline{x_2})$	$\overline{x_1 \vee x_2}$
0	0	1	1	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	0
1	0	1	1	0	0	1	1	0	0
1	1	1	0	1	0	1	0	1	0
		Eins	NAND	Implikation	Negation NOT	Implikation	Negation NOT	Äqivalenz	NOR

NAND und NOR

Übung

Übung

3.5 NAND und NOR

Mit NOT und AND (bzw. mit NAND) Gattern lassen sich alle anderen Gatter ersetzen. Gleiches gilt für NOT und OR bzw. NOR. Zeigen Sie dies, indem Sie entsprechende Schaltungen mit dem Simulationswerkzeug entwickeln:

a) Entwerfen Sie eine Schaltung aus NAND Gattern, welche für zwei Eingänge die OR-Verknüpfung als Ausgabe liefert.

b) Entwerfen Sie eine Schaltung aus NOR-Gattern, welche für zwei Eingänge die AND-Verknüpfung als Ausgabe liefert.

c) Wie müssten die Schaltungen erweitert werden, um für mehr als zwei Eingänge die AND bzw. OR Verknüpfung zu ergeben?

n-stufige Logik

Definition

Eine digitale Schaltung (Logik) wird als ***n-stufig*** bezeichnet, wenn zwischen Eingang und Ausgang n Gatterstufen in Kette geschaltet sind. Negation am Eingang oder Ausgang wird nicht als separate Stufe gezählt.

In realen Schaltungen benötigt das Signal ein gewisse Zeit, um durch die Gatter zu laufen. Die Verzögerungszeiten der einzelnen Stufen addieren sich.

Laufzeiten

Übung

Übung

3.6 Laufzeiten

Entwerfen Sie mit dem Simulator ein Schaltnetz, welches die Verzögerungszeiten der Gatterstufen anschaulich macht.

Systematik für beliebige Wahrheitstabellen?

Wie lassen sich für beliebige Wahrheitstabellen Schaltnetze systematisch entwickeln?

A	B	C	f(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Realisierung beliebiger Schaltfunktionen

Vorgehen:

- Definiere für jede Zeile mit f(...)=1 eine geeignete AND-Verknüpfung aller (eventuell negierten) Eingangsvariablen.
- Verknüpfe diese Terme mit OR zu f(...).

A	B	C	f(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$(\neg A \wedge \neg B \wedge \neg C)$$
$$\vee$$
$$(\neg A \wedge B \wedge C)$$
$$\vee$$
$$(A \wedge \neg B \wedge C)$$
$$\vee$$
$$(A \wedge B \wedge C)$$

Minterm / Maxterm

Definition

Ein **Minterm** ist die konjunktive (= AND) Verknüpfung aller Eingangsvariablen (in negierter oder nicht-negierter Form).

Ein **Maxterm** ist die disjunktive (= OR) Verknüpfung aller Eingangsvariablen (in negierter oder nicht-negierter Form).

Minterme, Beispiele für Eingabe A, B, C

$\neg A \wedge B \wedge C \qquad A \wedge \neg B \wedge \neg C \qquad A \wedge B \wedge C \qquad ...$

Maxterme, Beispiele für Eingabe A, B, C

$A \vee B \vee \neg C \qquad A \vee \neg B \vee C \qquad \neg A \vee \neg B \vee \neg C \quad ...$

Normalformen

Definition

Eine Schaltfunktion *f* ist in **disjunktiver Normalform**, wenn sie eine Disjunktion von Konjunktionen der (ggf. negierten) Schaltvariablen ist.

Eine Schaltfunktion *f* ist in **konjunktiver Normalform**, wenn sie eine Konjunktion von Disjunktionen der (ggf. negierten) Schaltvariablen ist.

Disjunktive Normalform, Beispiel:

$f = (\neg A \wedge B \wedge C) \vee (A \wedge \neg C \wedge \neg B) \vee D$

Konjunktive Normalform, Beispiel:

$f = (A \vee B) \wedge (\neg A \vee \neg C) \wedge D \wedge (C \vee \neg B)$

Kanonische disjunktive Normalform: Disjunktive Verknüpfung von Mintermen

Kanonische konjunktive Normalform: Konjunktive Verknüpfung von Maxtermen

→ d.h. es werden stets alle Schaltvariablen verknüpft.

Disjunktive Normalform

Beispiel

A	B	C	f(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$(\neg A \wedge \neg B \wedge \neg C)$

$\vee$

$(\neg A \wedge B \wedge C)$

$\vee$

$(A \wedge \neg B \wedge C)$

$\vee$

$(A \wedge B \wedge C)$

"gute" Minterme
d.h. f(...)=1

$f(A,B,C)=(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge C) \vee (A \wedge B \wedge C)$

Disjunktive Normalform

Übung

Übung

3.7 Disjunktive Normalform

Gegeben sei folgende Wahrheitstabelle:

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

a) Erstellen Sie die Schaltfunktion für *f* als disjunktive Normalform.

b) Entwerfen Sie ein entsprechendes Schaltnetz.

c) Wieviele Schaltstufen benötigt man maximal, um beliebige Wahrheitstabellen als Schaltnetz zu realisieren?

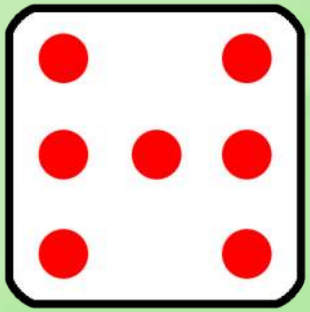
d) Optional: Wie könnte eine Systematik für die Entwicklung der *konjunktiven* Normalform aussehen?

Würfel

Übung

Übung

3.8 Würfel



Die Zustände eines Würfels lassen sich mit 3 Bit kodieren (z.B. Binärwerte für 0-5). Entwerfen Sie ein Schaltnetz, welches die sieben Lämpchen der abgebildeten digitalen Würfelanzeige ansteuert.

a) Erstellen Sie eine Wahrheitstabelle für die Schaltung.
b) Entwerfen (und simulieren) Sie ein geeignetes Schaltnetz.
c) Untersuchen Sie Schaltungsvarianten: Lässt sich die Anzahl der Gatter reduzieren? Lässt sich die Schaltung aus nur einem einzigen Gattertyp aufbauen?

Konjunktive Normalform

Beispiel

A	B	C	f(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$(A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (\neg A \vee B \vee C) \wedge (\neg A \vee \neg B \vee C)$$

$$f(A,B,C)=(A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (\neg A \vee B \vee C) \wedge (\neg A \vee \neg B \vee C)$$

Konjunktive Normalform

Übung

Übung

3.9 Konjunktive Normalform

Gegeben sei die Wahrheitstabelle aus Aufgabe 3.7.

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

a) Erstellen Sie die Schaltfunktion für *f* als konjunktive Normalform.
b) Entwerfen Sie ein entsprechendes Schaltnetz.
c) Vergleichen Sie die Ergebnisse des Schaltnetzes mit denen aus Aufgabe 3.7.
d) Wann würden Sie eher die konjunktive, wann die disjunktive Normalform verwenden?

Multiplikator

Übung

Übung

3.10 Multiplikator

Gegeben seien zwei Eingabewerte zwischen 0 und 3, die jeweils durch 2 Bit repräsentiert werden. Entwerfen Sie ein Schaltnetz, welches bzgl. des Produktes p beider Zahlen drei 1-Bit-Ausgaben liefert:

a1: p=0 a2: p>7 a3: p ist eine gerade Zahl

a) Erstellen Sie eine Wahrheitstabelle für die Schaltung.

b) Optional: Entwerfen die Schaltfunktionen in disjunktiver (optional: auch in konjunktiver) Normalform und simulieren Sie ein geeignetes Schaltnetz.

c) Optional: Verändern/erweitern Sie das Schaltnetz, so dass das 4-Bit-Produkt der Eingabewerte ausgegeben wird.

Theoreme der Schaltalgebra

Idempotenzen:	$A \wedge A = A$	$A \vee A = A$
Kommutativität:	$A \wedge B = B \wedge A$	$A \vee B = B \vee A$
Null-/Einselement:	$A \wedge 1 = A$ $\quad A \vee 1 = 1$	$A \wedge 0 = 0$ $\quad A \vee 0 = A$
Doppelte Negation:	$\neg\neg A = A$	
Assoziativität:	$(A \wedge B) \wedge C = A \wedge (B \wedge C)$	$(A \vee B) \vee C = A \vee (B \vee C)$
Absorption:	$A \wedge (A \vee B) = A$	$A \vee (A \wedge B) = A$
Distributivität:	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$	
De Morgan:	$\neg(A \wedge B \wedge \ldots \wedge X) = \neg A \vee \neg B \vee \ldots \vee \neg X$ $\neg(A \vee B \vee \ldots \vee X) = \neg A \wedge \neg B \wedge \ldots \wedge \neg X$	
Shannon:	$\neg f(A, B, C, \ldots, \wedge, \vee) = f(\neg A, \neg B, \neg C, \ldots, \vee, \wedge)$	

Theoreme der Schaltalgebra

Übung

Übung

3.11 Theoreme der Schaltalgebra

In der Vorlesung wurden verschiedene Rechenregeln der Schaltalgebra vorgestellt.

a) Veranschaulichen Sie die Rechenregeln mit Hilfe von Schaltnetzen (im Simulator), z.B. indem Sie für die rechte und linke Seite der Gleichungen Schaltnetze erstellen und die Ausgaben vergleichen.

b) Untersuchen Sie Ihre bisher entworfenen Schaltungen und ersetzen Sie Teile darin durch Anwendung der Schalttheoreme. Lassen sich die Schaltungen dadurch vereinfachen?

c) Optional: Beweisen Sie das De Morgan'sche Theorem für zwei Variablen mit Hilfe einer Wahrheitstabelle.

Normalform - Extrembeispiel

Beispiel

A	B	C	f(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge B \wedge \neg C)$$

Geht es auch einfacher?

Wie lassen sich mögliche Vereinfachungen systematisch erkennen?

Geht es auch einfacher?

Beispiel

A	B	C	f(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge B \wedge \neg C)$$

Geht es auch einfacher?

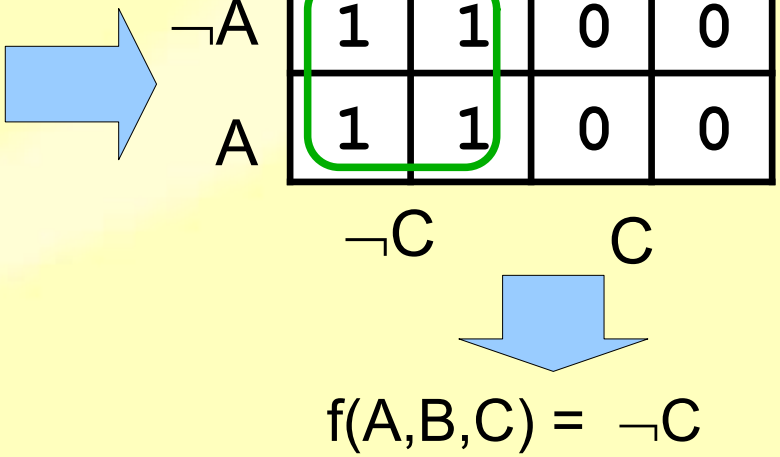
Wie lassen sich mögliche Vereinfachungen systematisch erkennen?

Karnaugh-Veitch-Diagramm

Beispiel

Beispiel 1

A	B	C	f(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	0	0

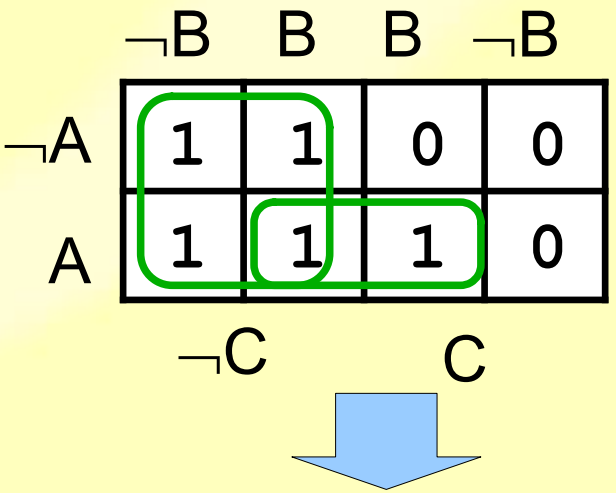


Karnaugh-Veitch-Diagramm

Beispiel

Beispiel 2

A	B	C	f(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



$$f(A,B,C) = \neg C \vee (A \wedge B)$$

KV-Diagramm

Übung

Übung

3.12 KV-Diagramm

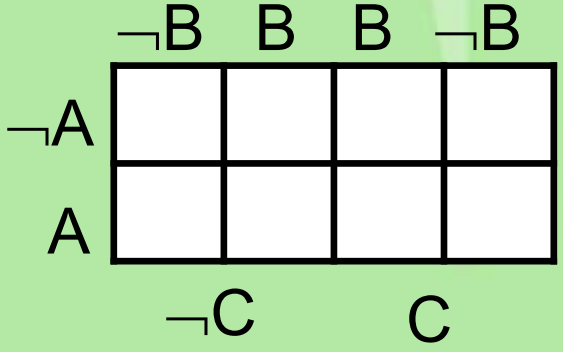
Gegeben sei die folgende Wahrheitstabelle:

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

a) Erstellen Sie die Schaltfunktion als disjunktive Normalform.

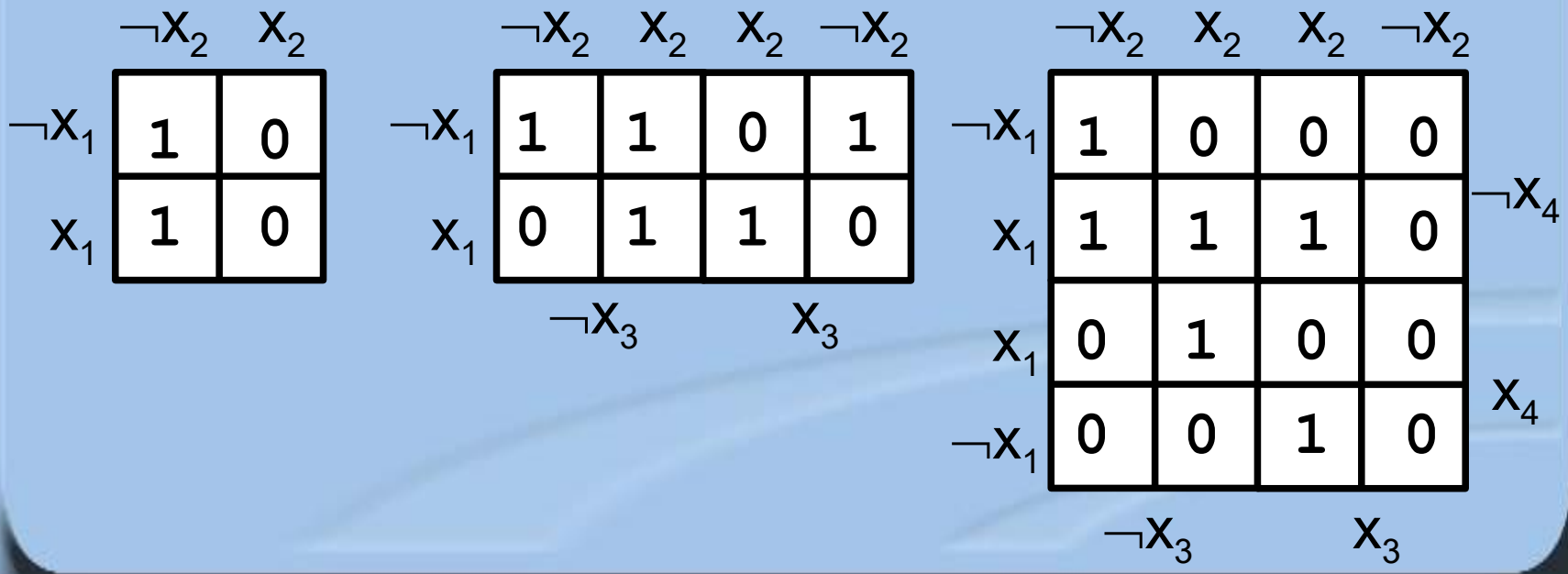
b) Erstellen Sie das KV-Diagramm.

c) Geben Sie eine vereinfachte Schaltfunktion an.



KV-Diagramme für 2-4 Variablen

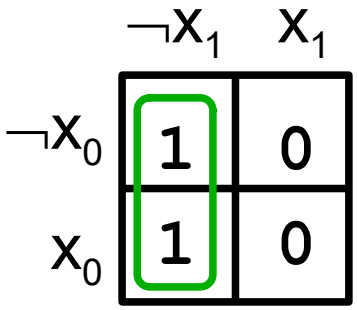
Karnaugh-Veitch-Diagramm (kurz KV-Diagramm)

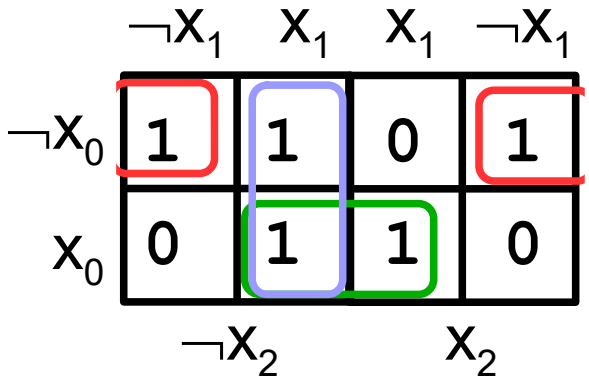


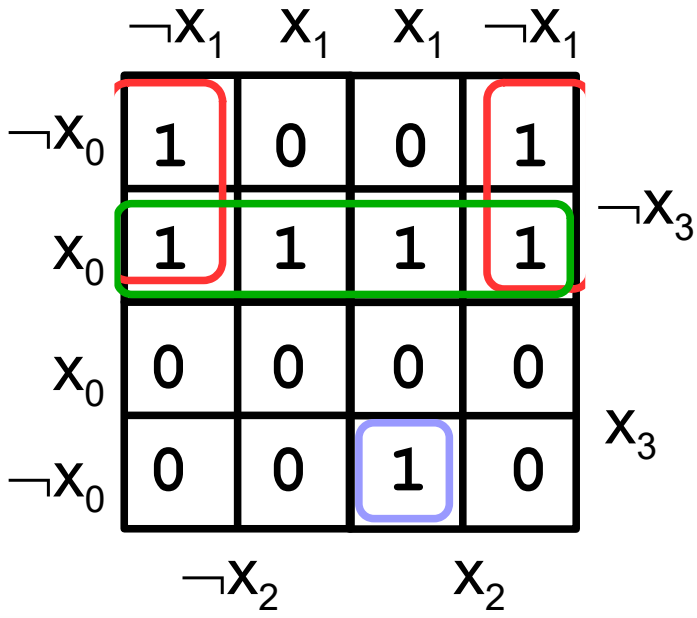
Vereinfachung von Schaltfunktionen

Zur Vereinfachung einer Schaltfunktion werden im KV-Diagramm rechteckige Blöcke gebildet, die 2^k benachbarte Felder mit Wert 1 zusammenfassen (auch die gegenüber liegenden Randfelder gelten als benachbart). Jedes Feld mit einer 1 muss in einem Block enthalten sein.

Jeder Block entspricht einer Konjunktion, ihre disjunktive Verknüpfung ist die vereinfachte Schaltfunktion. Die den größtmöglichen Blöcken entsprechenden Terme bezeichnet man als **Primterme**.







KV-Diagramm für Multiplikator

Übung

Übung

3.13 KV-Diagramm für Multiplikator

Betrachten Sie den Multiplikator aus Aufgabe 3.10: Gegeben seien zwei Eingabewerte zwischen 0 und 3, die jeweils durch 2 Bit repräsentiert werden. p sei das Ergebnis der Multiplikation beider Werte.

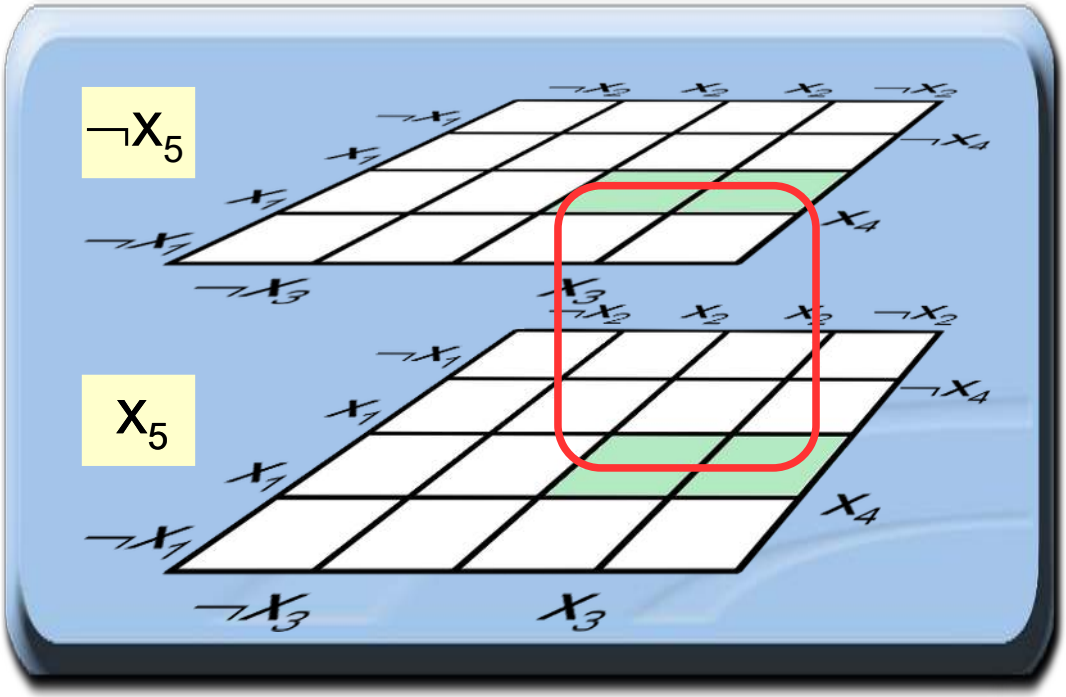
a) Erstellen Sie eine Wahrheitstabelle für das zweite Bit des Ergebnisses der Multiplikation p.

b) Erstellen Sie das entsprechende KV-Diagramm und geben Sie eine vereinfachte Schaltfunktion an.

c) Optional, zusätzlich: Entwerfen Sie mit KV-Diagrammen Schaltfunktionen für drei 1-Bit-Ausgaben bzgl. des Produktes p:
   a1: p=0 a2: p>7 a3: p ist eine gerade Zahl

d) Wieviele Gatter und Eingänge sparen Sie durch die Vereinfachung im Vergleich zu einer Realisierung entsprechend der disjunktiven Normalform?

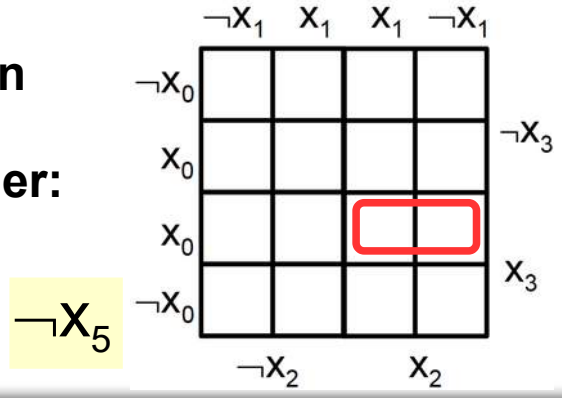
KV-Diagramm für 5 Variablen

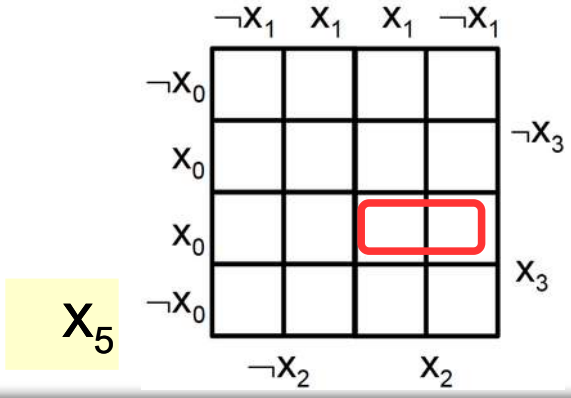


Für 6 Variablen:

2 zusätzliche Ebenen

Oder Notation als Blöcke nebeneinander:



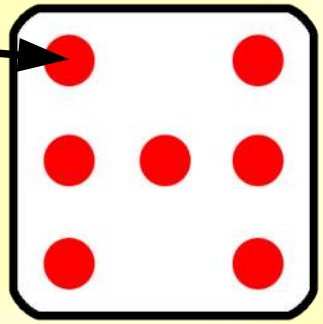


Unvollständige Wahrheitstabellen

Beispiel: Würfel

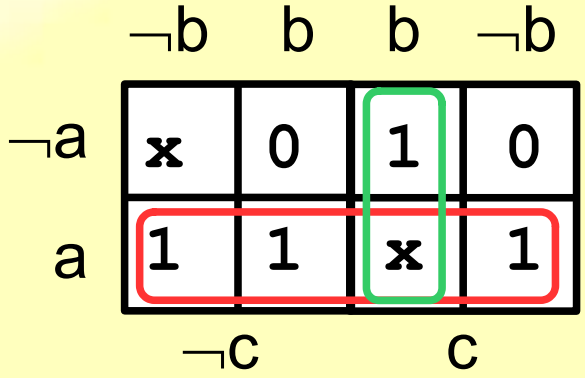
nur die Eingabewerte 1 bis 6 sind relevant

a	b	c	f(a,b,c)
0	0	0	x
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	x



don't care / egal

"gleichgültige Minterme:

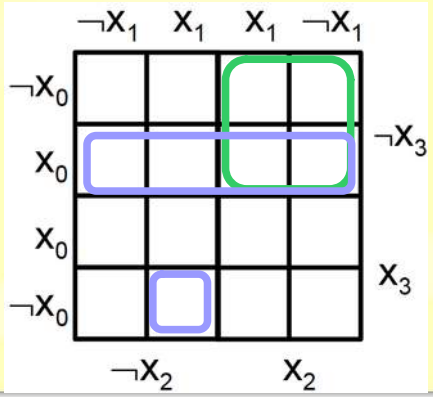


Mehrere Ausgänge

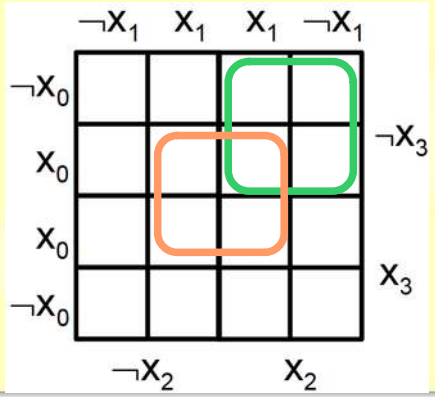
Bei Schaltungen mit mehreren Ausgängen muss pro Ausgang ein KV-Diagramm erstellt werden.

Gibt es gleiche Blöcke in den verschiedenen Diagrammen, so kann der entsprechende Teil der Schaltung für die zugehörigen Ausgänge mehrfach verwendet werden und die Schaltung vereinfacht sich. U.u. lässt sich die Schaltung weiter vereinfachen, wenn größere Blöcke in kleinere, besser wiederverwendbare aufgeteilt werden.

Ausgang 1



Ausgang 2



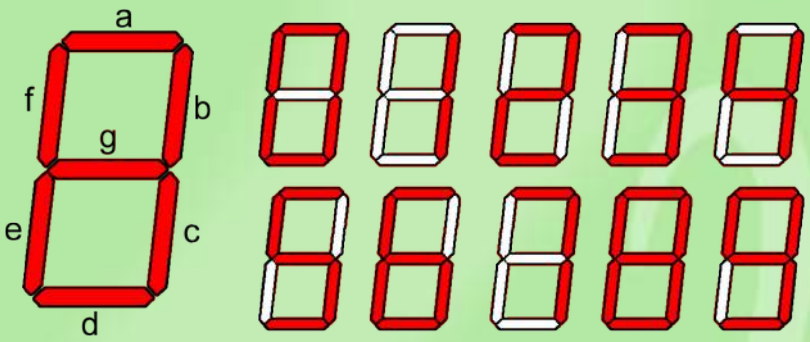
Der Term $x_2 \wedge \neg x_3$ muss in der Schaltung nur einmal realisiert werden.

Siebensegmentanzeige

Übung

Übung

3.14 Siebensegmentanzeige



Ein 4 Bit Wert zwischen 0 und 9 soll auf einer Siebensegmentanzeige ausgegeben werden. Hierzu müssen die Segmente a-g entsprechend des Eingangswertes angesteuert werden.

a) Erstellen Sie die Wahrheitstabellen für die Segmente a, c und e.

b) Erstellen Sie die zugehörigen KV-Diagramme und geben Sie die vereinfachten Schaltfunktionen an. Achten Sie auf den korrekten Umgang mit Don't Care Termen.

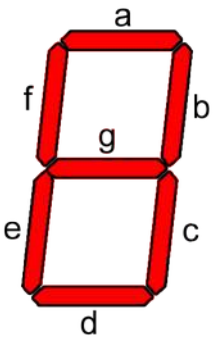
c) Welche Terme bzw. Teilschaltungen lassen sich für die Ansteuerung verschiedener Segmente wiederverwerten? Ist eine weitere Vereinfachung durch Aufteilen von Blöcken möglich?

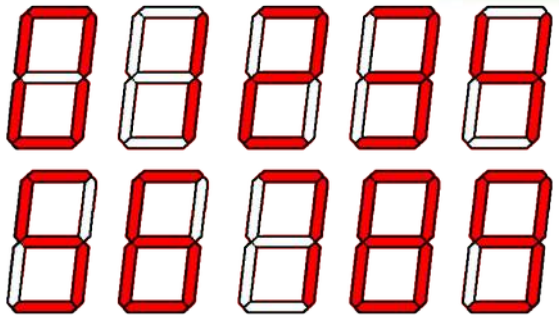
d) Optional: Führen Sie die obigen Schritte für die übrigen Segmente durch. Erstellen Sie die vollständige Schaltung.

Siebensegmentanzeige

Übung

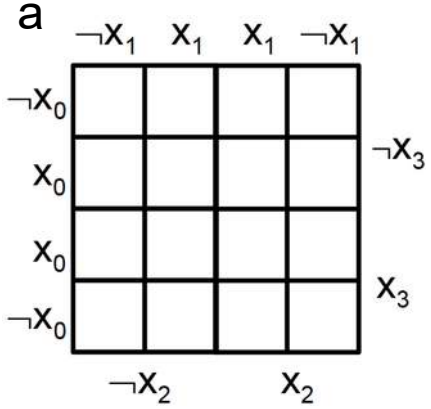
x_0	x_1	x_2	x_3	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x



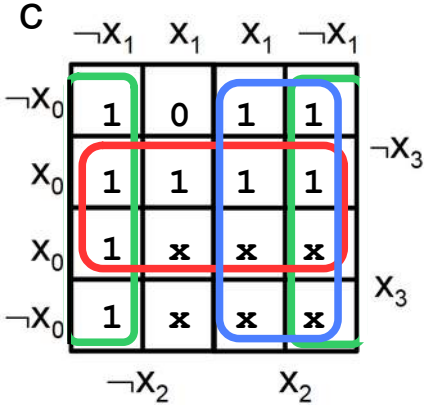


(!x2 & !x0) v (!x0 & x1)

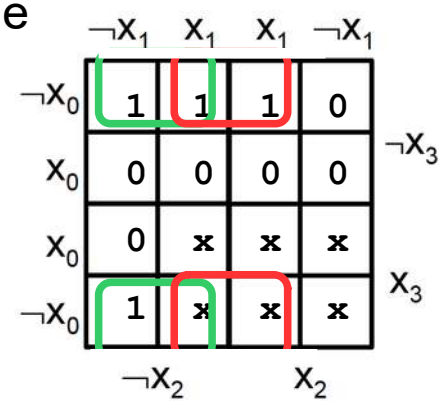
a



c



e



Glitch / Hazard

Definition

Glitch / Hazard: Bei einem Wechsel der Eingabe ist die Ausgabe vor Erreichen des korrekten Endzustandes kurzzeitig falsch. Ursache sind Laufzeitunterschiede (→ *Race condition*) der verschiedenen Wege des Signals durch das Schaltnetz.

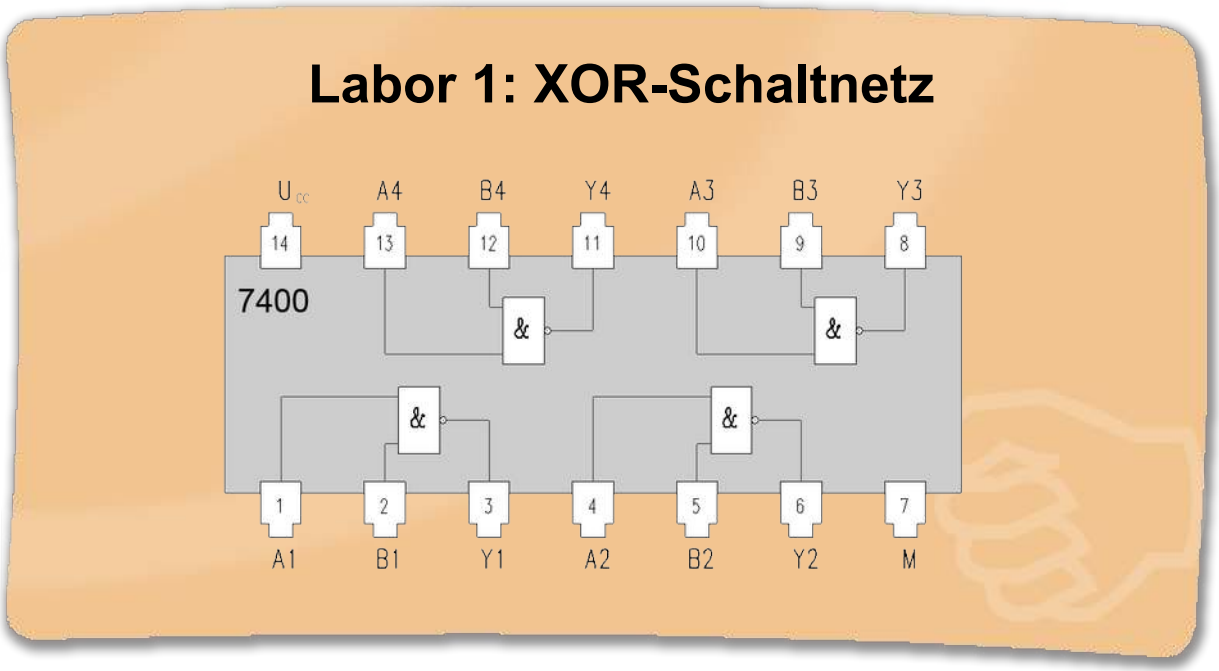
Kategorien:

- Statische / dynamische Glitches
  - Ausgabe: Anfangswert ist gleich / ungleich Endwert
- 0-Glitch / 1-Glitch
  - gewünschter Ausgabewert ist 0 / 1
- Struktur-Glitch
  - s. KV-Diagram-Block-Wechsel
  - → zusätzliche Primterme
- Funktions-Glitch
  - mehrere Eingänge wechseln
  - → ggf. Gray-Code verwenden

Welche Gatter in der letzten Schaltstufe können zu statischen 0-Glitches führen, welche zu statischen 1-Glitches?

Experimentieren Sie mit dem Pegelschreiber des Digitalsimulators

Labor 1: XOR-Schaltnetz



Labor: Addierer

Labor

2. Halbaddierer

Halbaddierer: Entwerfen Sie eine Schaltung, die zwei 1-Bit-Eingangswerte A und B so verknüpft, dass sie als Ergebnis zwei Bit Ü und S ergibt, die die binäre Summe (bzw. die 1-Bit Summe und den Übertrag) der Eingangswerte darstellen.

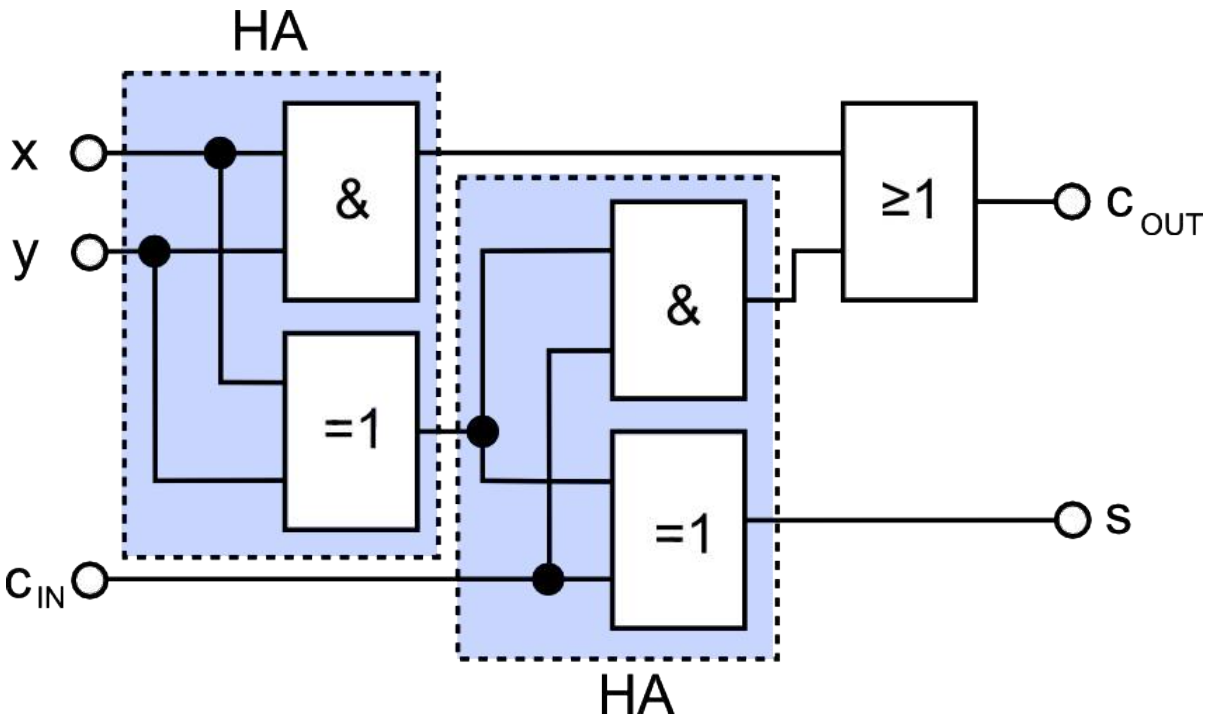
a) Erstellen Sie die Wahrheitstabelle und die Schaltfunktionen für Ü und S.
b) Bauen Sie die Schaltungen mit den Grundelementen AND, OR und NOT auf.
c) Testen Sie die Schaltung und notieren Sie die Ergebnisse in einer Wahrheitstabelle.

3. Volladdierer

Volladdierer: Erweitern Sie den Halbaddierer, so dass er bei der Eingabe zusätzlich ein Übertragsbit $Ü_{n-1}$ mit aufaddiert.

d) Erstellen Sie die Wahrheitstabelle für die Ausgabewerte $Ü_n$ und S.
e) Überlegen Sie, wie sich ein solcher Volladdierer aus zwei Halbaddierern aufbauen lässt.
f) Bauen Sie eine entsprechende Schaltung auf.
g) Testen Sie die Schaltung und notieren Sie die Ergebnisse und Zwischenwerte in einer Wahrheitstabelle.
h) Überlegen Sie, wie man aus mehreren Volladdierern einen Addierer für eine n-stellige Binärzahl aufbauen kann.

Labor 2: Addierer



n-stelliger Addierer