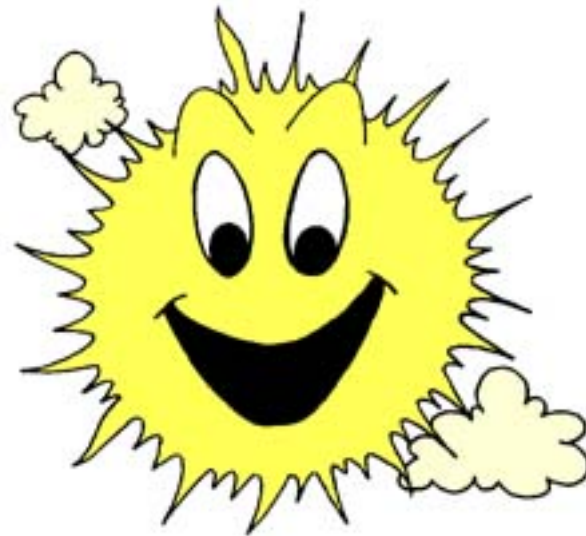


# Wiederholung der 3. Vorlesung



## 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.	$\neg(A \wedge B) = \neg A \vee \neg B$	$\neg(A \vee B) = \neg A \wedge \neg B$

## 3.5 Hörsaalübung Morgan'sche Gesetze

### Anwendung der Morganschen Gesetze auf logische Aussagen

- „Wenn morgen schönes Wetter ist und mein Bruder Zeit hat, gehen wir segeln.“
- „wenn **kein** schönes Wetter ist“ **oder** „mein Bruder **keine** Zeit hat“ „gehen wir **nicht** segeln“
- „Wenn ich eine Erbschaft mache oder im Lotto gewinne, mache ich eine Weltreise.“
- „**keine** Erbschaft“ **und** „**kein** Lottogewinn“ dann auch „**keine** Weltreise“

## 2.4.7 Beispiel zur 1. Kürzungsregel

$$X_1 \vee (X_1 \wedge X_2) \quad \text{Null-Eins-Gesetz} \\ = (X_1 \wedge 1) \vee (X_1 \wedge X_2)$$

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

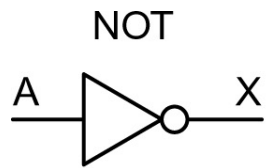
$$\quad \text{Null-Eins-Gesetz} \\ = X_1 \wedge 1$$

$$\quad \text{Null-Eins-Gesetz} \\ = X_1$$

## 2.4.7 Beispiel Umformung

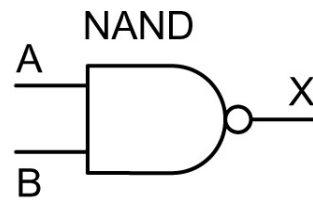
$$\begin{aligned} & ((A \vee \overline{(B \wedge A)})) \wedge (C \vee (D \vee C)) \\ &= ((A \vee (\overline{B} \vee \overline{A}))) \wedge (C \vee (D \vee C)) \\ &= (((A \vee \overline{A}) \vee \overline{B})) \wedge ((C \vee C) \vee D) \\ &= (1 \vee \overline{B}) \wedge (C \vee D) \\ &= 1 \wedge (C \vee D) \\ &= (C \vee D) \end{aligned}$$

## 3.2 Schaltzeichen nach ANSI / ISO



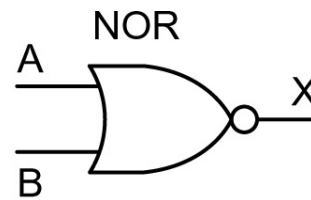
A	X
0	1
1	0

(a)



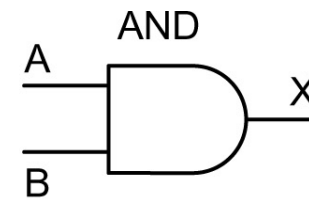
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)



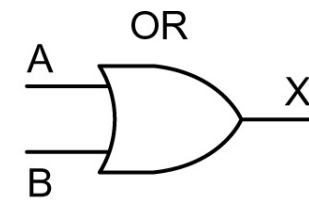
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

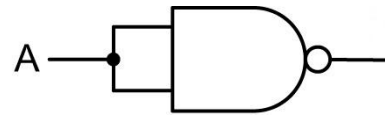


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

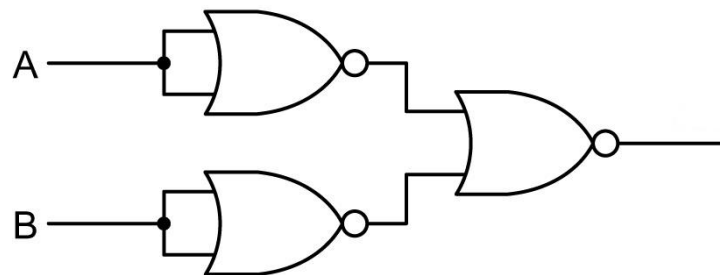
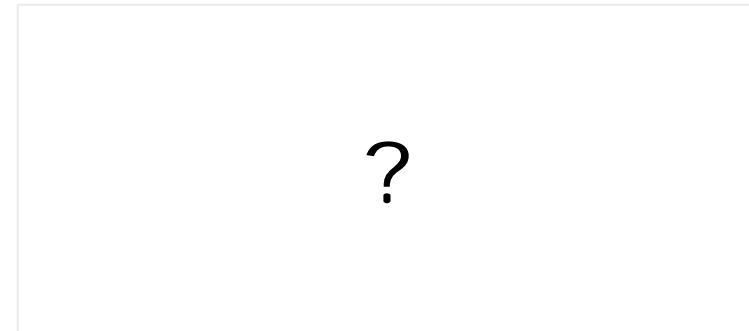
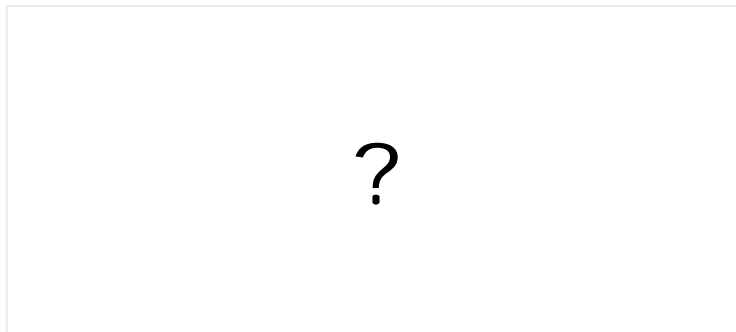
(e)

## 3.2 Schaltungs-Äquivalenz (HS-Übung 1)

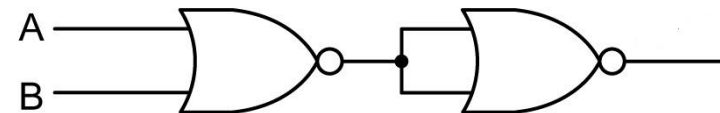
für (a)-(c)  
nur NAND  
und NOR  
verwenden!



(a)

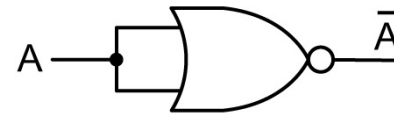
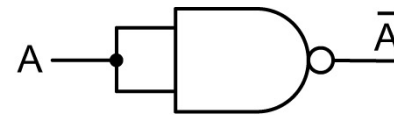


(b)

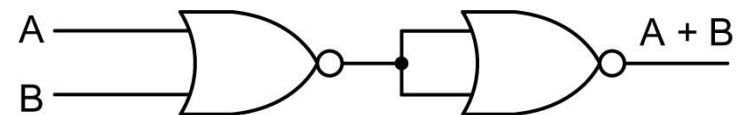
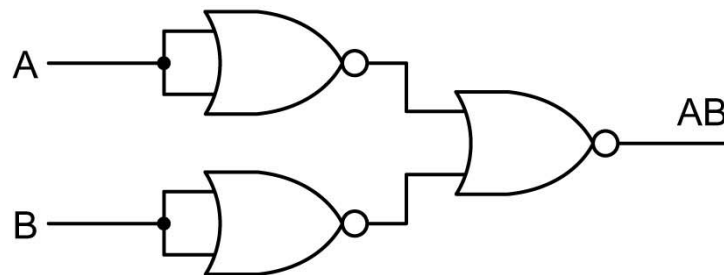
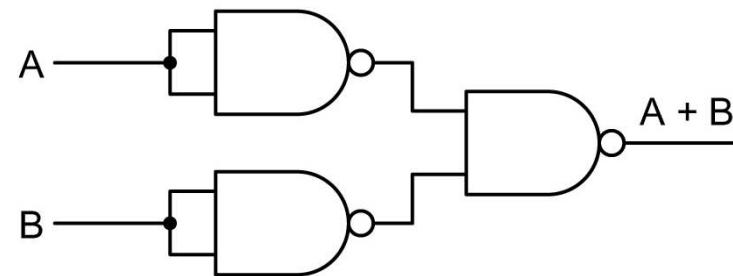
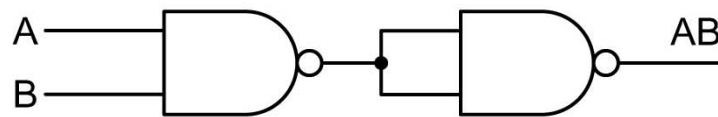


(c)

## 3.2 Schaltungs-Äquivalenz (Lösung von 1)



(a)

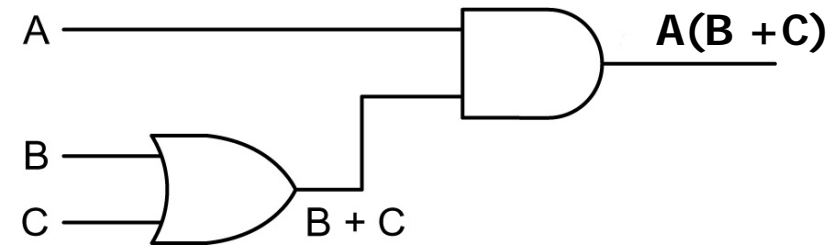
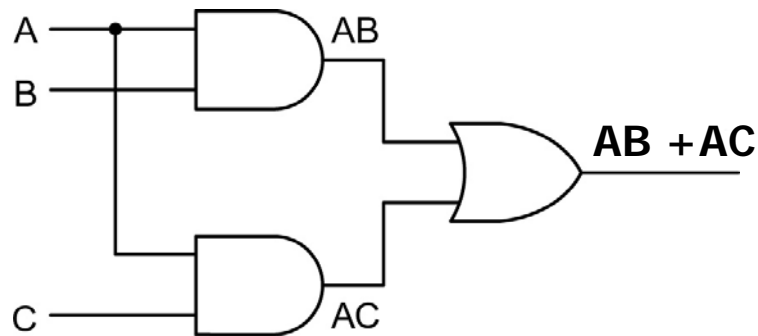


(b)

(c)

- Was macht die folgende Schaltung?
- Wahrheitstabelle!
- Vereinfachen Sie die Schaltung mit AND und OR!
- Überprüfen Sie mit Wahrheitstabelle
- Welches Gesetz wird hier angewendet?

## 3.2 Schaltungs-Äquivalenz (HS-Übung 2)



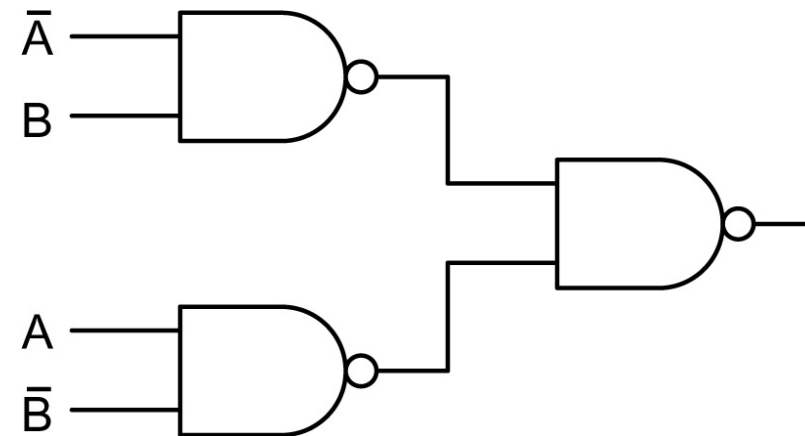
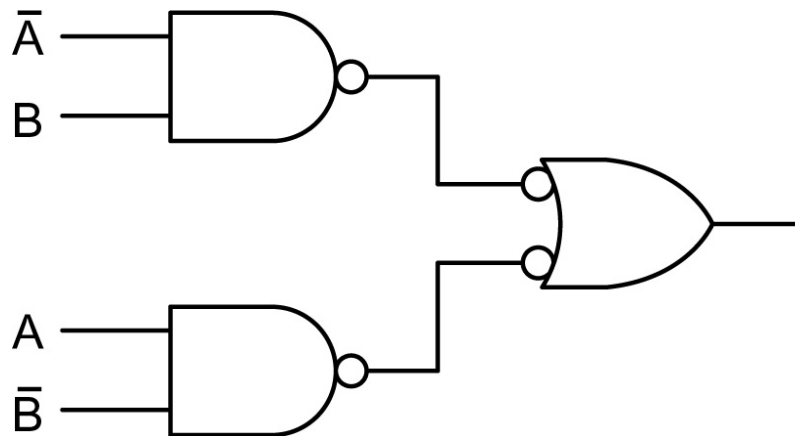
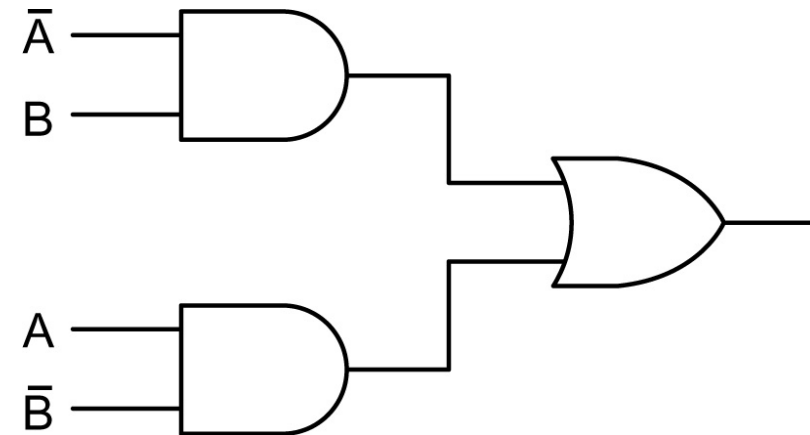
A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

A	B	C	A	B + C	A(B + C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

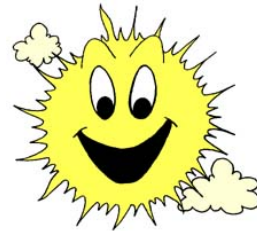
## 3.2 Schaltungs-Äquivalenz (HS-Übung 3)

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Realisieren sie die Schaltung für die gegebene Wahrheitstabelle; verwenden sie als Variablen  $A, \bar{A}, B, \bar{B}$



# Ende der Wiederholung

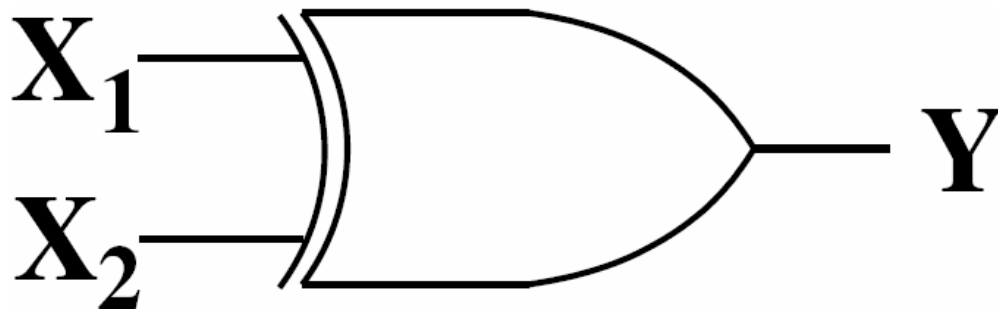
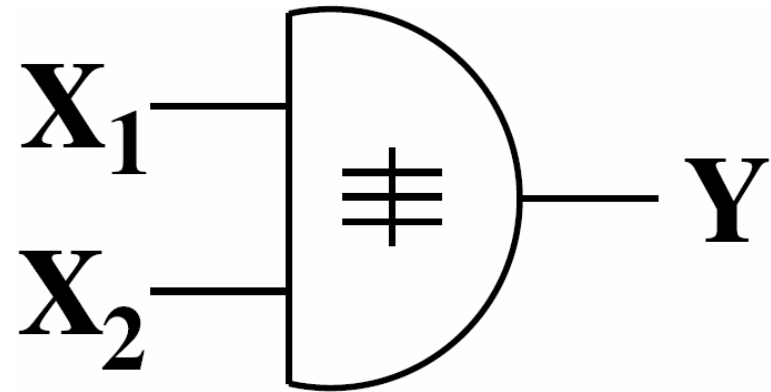
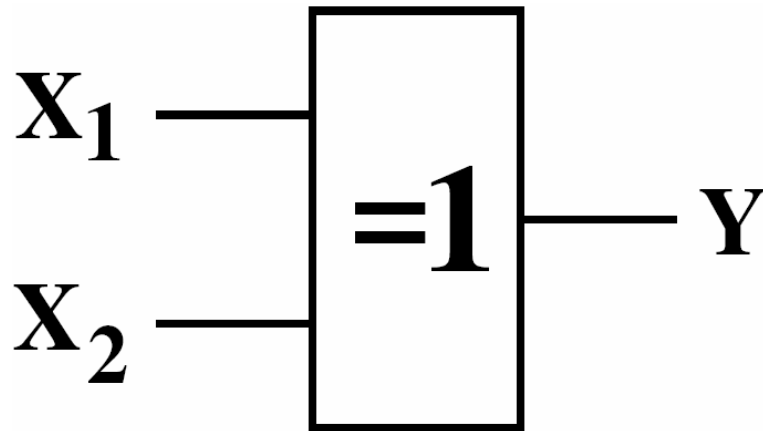


## 3.2. XOR (Antivalenz)

XOR (exclusive OR) ist die Realisierung des „Entweder-oder“

$X_1$	$X_2$	$X_1 \wedge \overline{X_2}$	$\overline{X_1} \wedge X_2$	$(X_1 \wedge \overline{X_2}) \vee (\overline{X_1} \wedge X_2)$
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

## 3.2. XOR Symbole



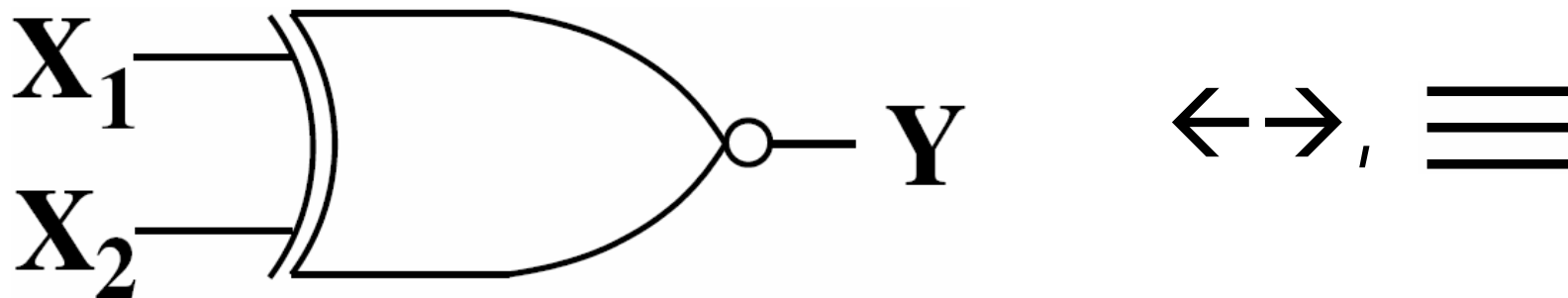
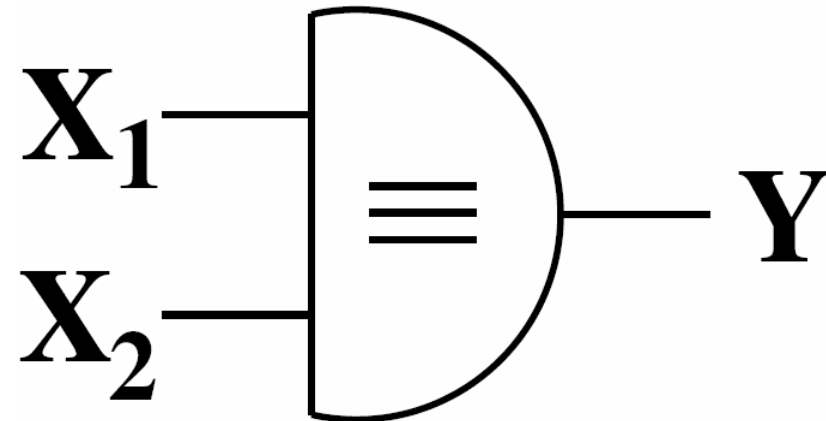
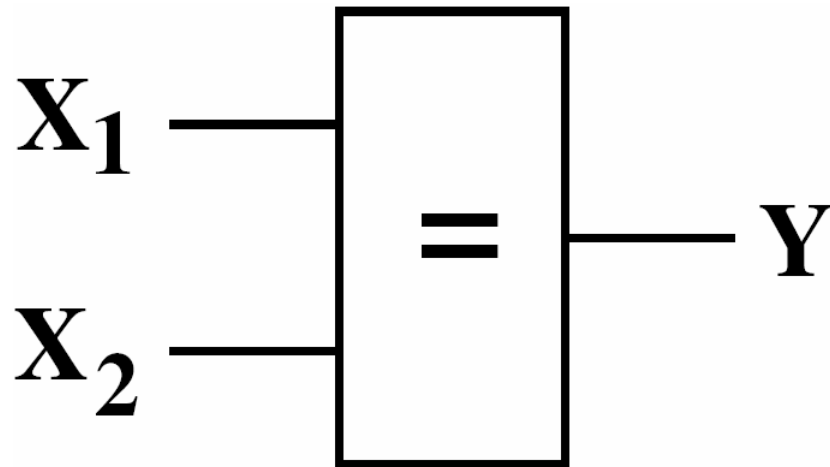
$\oplus$ ,  $\neq$

## 3.2. XNOR (Äquivalenz)

XNOR (exclusive NOR) überprüft die Gleichheit (Äquivalenz) zweier Variablen.

$X_1$	$X_2$	$X_1 \wedge X_2$	$\overline{X_1} \wedge \overline{X_2}$	$(X_1 \wedge X_2) \vee (\overline{X_1} \wedge \overline{X_2})$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

## 3.2. XNOR Symbole



## 5 Vollst. Systeme mit NAND und NOR

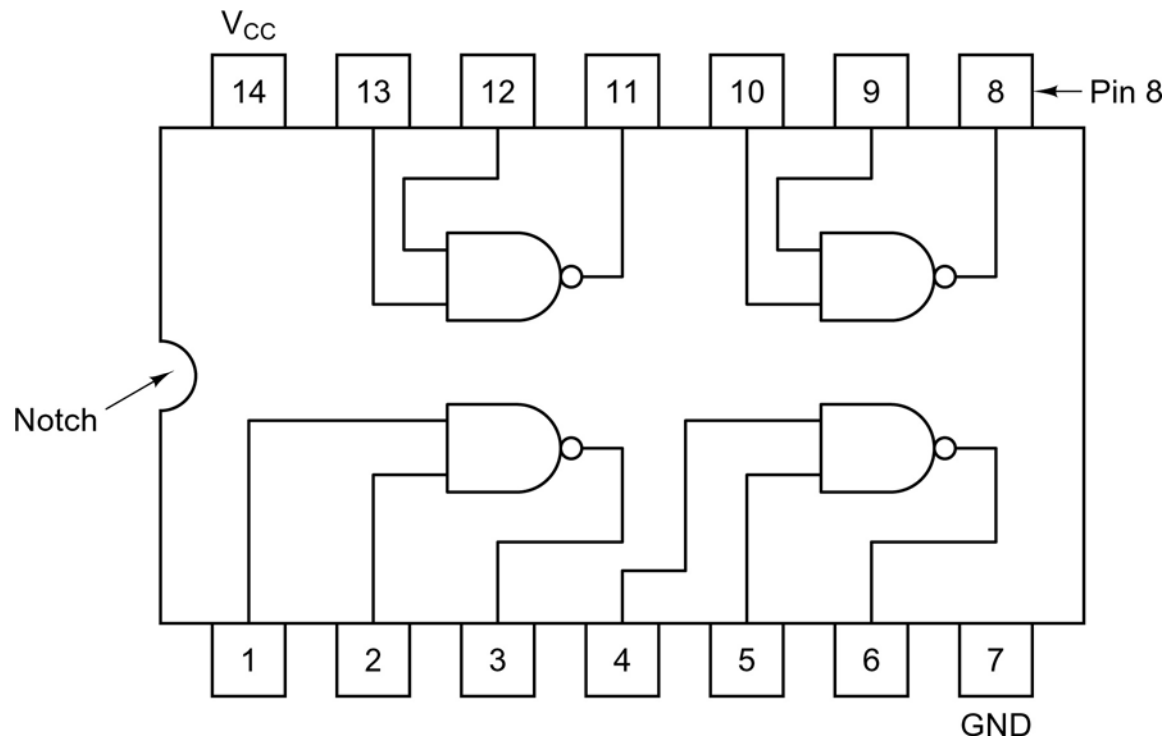
1. Normalformen
2. Minterm (Vollkonjunktion)
3. Maxterme (Volldisjunktion)
4. Disjunktive Normalform (DNF)
5. Konjunktive Normalform (KNF)
6. Minimierungsverfahren
7. KV-Diagramme
8. Minimierungsverfahren nach Quine und Mc Cluskey
9. Binäre Entscheidungsdiagramme

Ebenso wie die AND-, OR-Verknüpfung und NOT bilden NAND und NOR, jede für sich, ebenfalls vollständige Systeme.

Somit reicht es für die Entwicklung einer booleschen Hardware aus, entweder eine Grundschialtung für NAND oder für NOR zur Verfügung zu stellen.

Daraus kann durch Replizieren und Kombinieren jede boolesche Schaltung entwickelt werden.

## 5 Beispiel eines integrierten Bausteins (IC)



- SSI (Small Scale Integrated): 1-10 Gates
- MSI (Medium Scale Integrated): 10-100 Gates
- LSI (Large Scale Integrated): 100-100000 Gates
- VLSI (Very Large Scale Integrated): >100000 Gates

## 5.x Addition und Subtraktion (s.a. RG)

Addition:

$$0 + 0 = 0$$

$$1 + 0 =$$

$$0 + 1 = 1, \text{ kein Übertrag (Carry)}$$

$$1 + 1 = 0, \text{ Übertrag } 1$$

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	0	1
1	1		

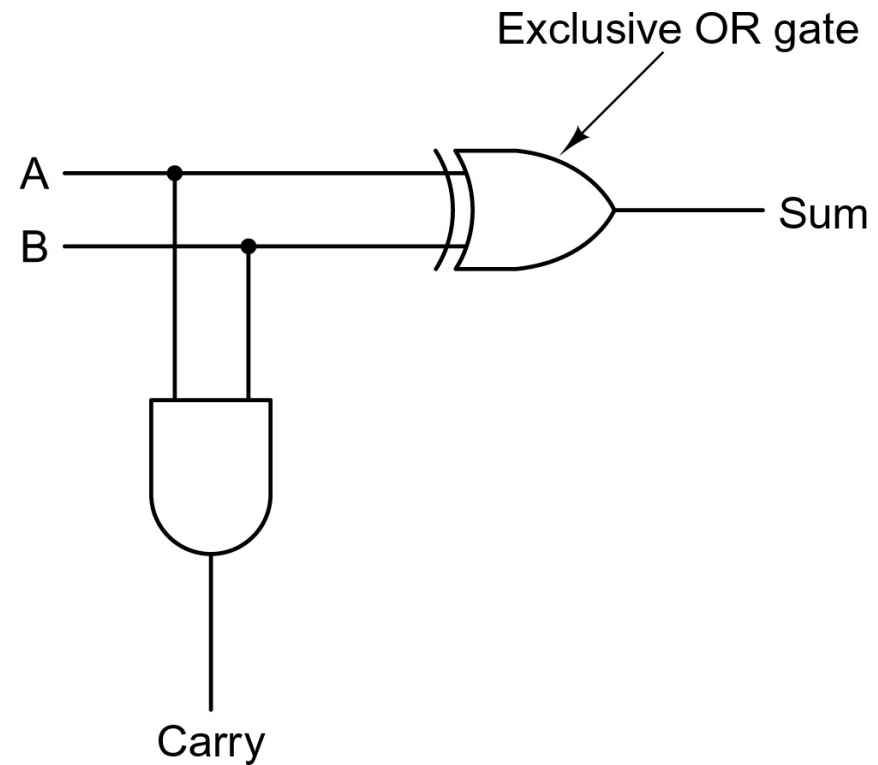
$$\begin{aligned} Sum &= A \oplus B \\ &= (\bar{A} \wedge B) \vee (A \wedge \bar{B}) \end{aligned}$$

$$Carry_{out} = A \wedge B$$

# 5.x Halbaddierer (Half Adder)

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{array}{r} 6_{10} \quad 0110 \\ + 7_{10} \quad 0111 \\ \hline \text{C} \quad 110 \\ \hline = 13_{10} \quad 1101 \end{array}$$



Kann Addition mit Halbaddierer realisiert werden?

Fehlt etwas?

- Fortsetzung Addierer, Subtrahierer, Multiplikation und Division binärer Zahlen in Vorlesung Rechnergrundlagen

## Digitaltechnik:

- Normalformen
- ...