# Boolesche Algebra (1)

**Definition 1:**

Sei $B = \Sigma_2 = \{0,1\}$ das Alphabet mit den Elementen 0 und 1. Seien auf B die 3 Operatoren einer Algebra wie folgt definiert für x,y aus B:

$x+y := \mathrm{Max}(x,y)$, $x \cdot y := \mathrm{Min}(x,y)$, $\neg x :=$ Komplement von x

Dann ist $(B,+,\cdot,\neg)$ eine *Boolesche Algebra* (d.h. Algebra über einem 2-elementigen Alphabet).

**Satz 1:**

In dieser Algebra gelten das

- Kommutativgesetz: $\mathbf{x+y = y+x, xy = yx}$
- Assoziativgesetz: $\mathbf{(x+y)+z = x+(y+z), xy(z) = x(yz)}$
- Distributivgesetz: $\mathbf{x(y+z) = xy+xz, x+(yz) = (x+y)(x+z)}$
- Komplementgesetz: $\mathbf{x+\neg x = 1, x\neg x = 0}$
- Idempotenzgesetz: $\mathbf{x+x = x, xx = x, \neg\neg x = x}$
- Gesetz vom kleinsten und größten Element: $\mathbf{x+0 = x}$ und $\mathbf{x+1 = 1, x0 = 0}$ und $\mathbf{x1 = x}$
- de Morganschen Regeln: $\mathbf{\neg(x+y) = \neg x\neg y, \neg(xy) = \neg x+\neg y}$

**---> Prinzip der Dualität:**

Sei *p* ein Axiom oder aus den Axiomen der BA abgeleiteter Satz, dann gilt auch der zu *p* duale Satz *p‘*.

**---> Negation ist nicht distributiv, sondern statt dessen gilt de Morgan!**

**---> Subtraktion und Division sind in der BA nicht definiert!**

# Boolesche Algebra (2)

**Definition 2:**

Seien *n,m* aus **N**, *n,m* $\geq 1$. Dann heißt eine Funktion **F**: $B^n \dashrightarrow B^m$ eine *Schaltfunktion* in *n* Variablen mit *m* Ausgängen.

**Definition 3:**

Eine Schaltfunktion **f**: $B^n \dashrightarrow B$ heißt (n-stellige) *Boolesche Funktion.* $B_{n,1}$ bzw.abkürzend $B_n$ bezeichnet die Menge aller Booleschen Funktionen mit n Variablen. $ON(f) := \{ x \in B_n , f(x) = 1\}$ heißt die *Erfüllbarkeitsmenge* von f.

**Satz 2:**

Eine totale (vollständig definierte) Boolesche Funktion ist bereits allein durch ihre Erfüllbarkeitsmenge eindeutig bestimmt.

**Satz 3:** $(B_n, +, \cdot, \neg)$ ist eine Boolesche Algebra.

**Definition 4:** Ein Boolescher Ausdruck der Form $x_i^j$ (j = 0,1) heißt *Literal.* Dabei ist $x_1^1 := x_1$ und $x_1^0 := \neg x_1$.

**Definition 5:**

Ein *Monom (Term)* ist eine Konjunktion (Produkt) von Literalen, in der kein Literal doppelt oder gleichzeitig in seiner positiven und negativen Variante vorkommt. Ein Monom *m* heißt *vollständig,* wenn jede Variable $x_i$ entweder als positives oder negatives Literal in m vorkommt.

# Boolesche Algebra (2a)

**Definition 6:**
Ein Boolescher Ausdruck (BA) bestehend aus einer Disjunktion (Summe) paarweise verschiedener Monome heißt *Polynom.*

**Satz 5:** Zu jeder Booleschen Funktion existieren beliebig viele Boolesche Ausdrücke

**Definition 7:**
Sei **f**: $B^n$ --> B eine Boolesche Funktion (BF) und sei $x_i = (x_1^j, x_2^j, \ldots, x_n^j)$ aus $B^n$ . Dann heißt das Produkt $x_1^j x_2^j \ldots x_n^j$ der Elemente von $x_i$ ein *Minterm* von **f**. Dieser heißt *einschlägig*, wenn **f** $(x_1, x_2, \ldots, x_n) = 1$, d.h. wenn $x_i$ ein Element der Erfüllbarkeitsmenge von f ist.

**Satz 6:** Die Begriffe vollständiges Monom und Minterm sind äquivalent.

**Definition 8:**
Die Darstellung einer BF durch einschlägige Minterme heißt auch *disjunktive Normalform (DNF)* einer BF.

**Satz 7:** Jede BF ist eindeutig darstellbar als Summe ihrer einschlägigen Minterme.

**Korollar 1:**
Jede n-stellige BF ist allein darstellbar durch die 2-stelligen BF + und · sowie die 1-stellige BF ¬. Anders ausgedrückt: {+, ·, ¬} ist *funktional vollständig.*

**Korollar 2:** {+, ¬} und {·, ¬} sind funktional vollständig.

**Korollar 3:** {NAND} bzw {NOR} sind für sich allein funktional vollständig.

# Boolesche Algebra (2b)

**Gesetze der Booleschen Algebra(BA):**

- Kommutativgesetz: **x+y = y+x, xy = yx**
- Assoziativgesetz: **(x+y)+z = x+(y+z), xy(z) = x(yz)**
- Distributivgesetz: **x(y+z) = xy+xz, x+(yz) = (x+y)(x+z)**
- Komplementgesetz: **x+¬x = 1, x¬x = 0**
- Idempotenzgesetz: **x+x = x, xx = x, ¬¬x = x**
- das Gesetz vom kleinsten und größten Element: **x+0 =x** und **x+1 = 1, x0 = 0** und **x1 = x**
- die der Morganschen Regeln: **¬(x+y) = ¬x¬y, ¬(xy) = ¬x+¬y**

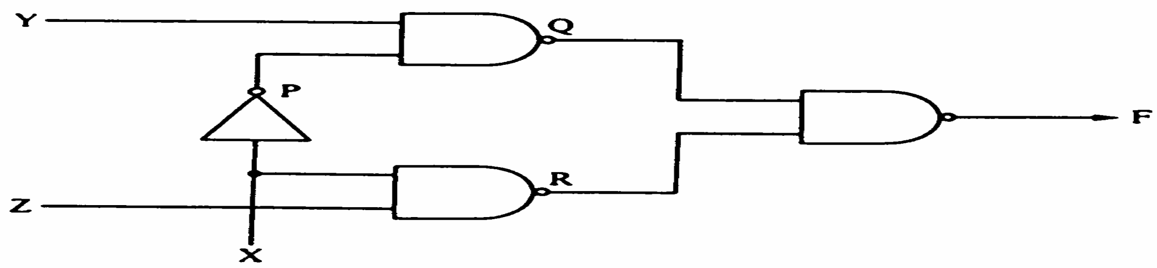
**Beispiele für abgeleitete Sätze:**

Satz 8: x + xy = x (Absorptionsgesetz)
Satz 9: x + ¬xy = x + y
Satz 10: xy + ¬xz + yz = xy + ¬xz
Satz 11: (x + y)(¬x +z)(y + z) = (x + y)(¬x +z) (dualer Satz zu Satz 10)

# Boolesche Algebra (3a)

**Beispiel Multiplexer:**

el 2:

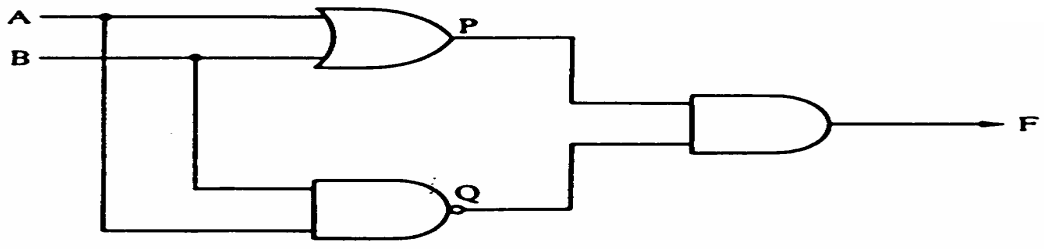


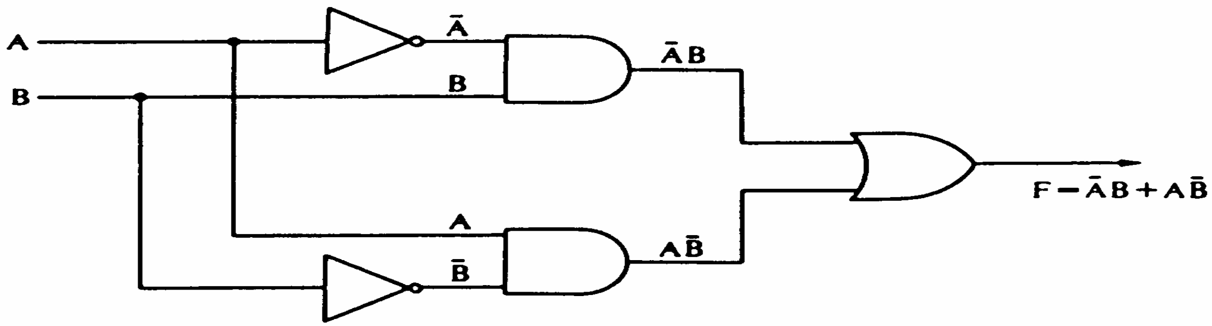

he

| Inputs | | | Intermediate values | | | Output |
|---|---|---|---|---|---|---|
| X | Y | Z | $P=\overline{X}$ | $Q=\overline{YP}$ | $R=\overline{XZ}$ | $F=\overline{QR}$ |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 |

# Boolesche Algebra (3b)

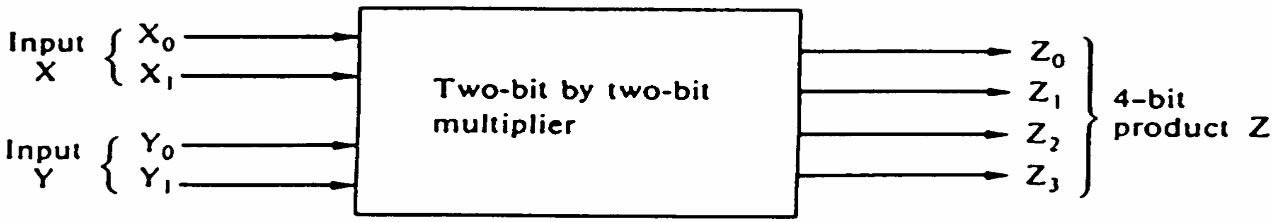
**Äquivalente Schaltnetze für EXOR:**



# Boolesche Algebra (3)

**Beispiel: Entwurf eines 2-bit - Multiplizierers**

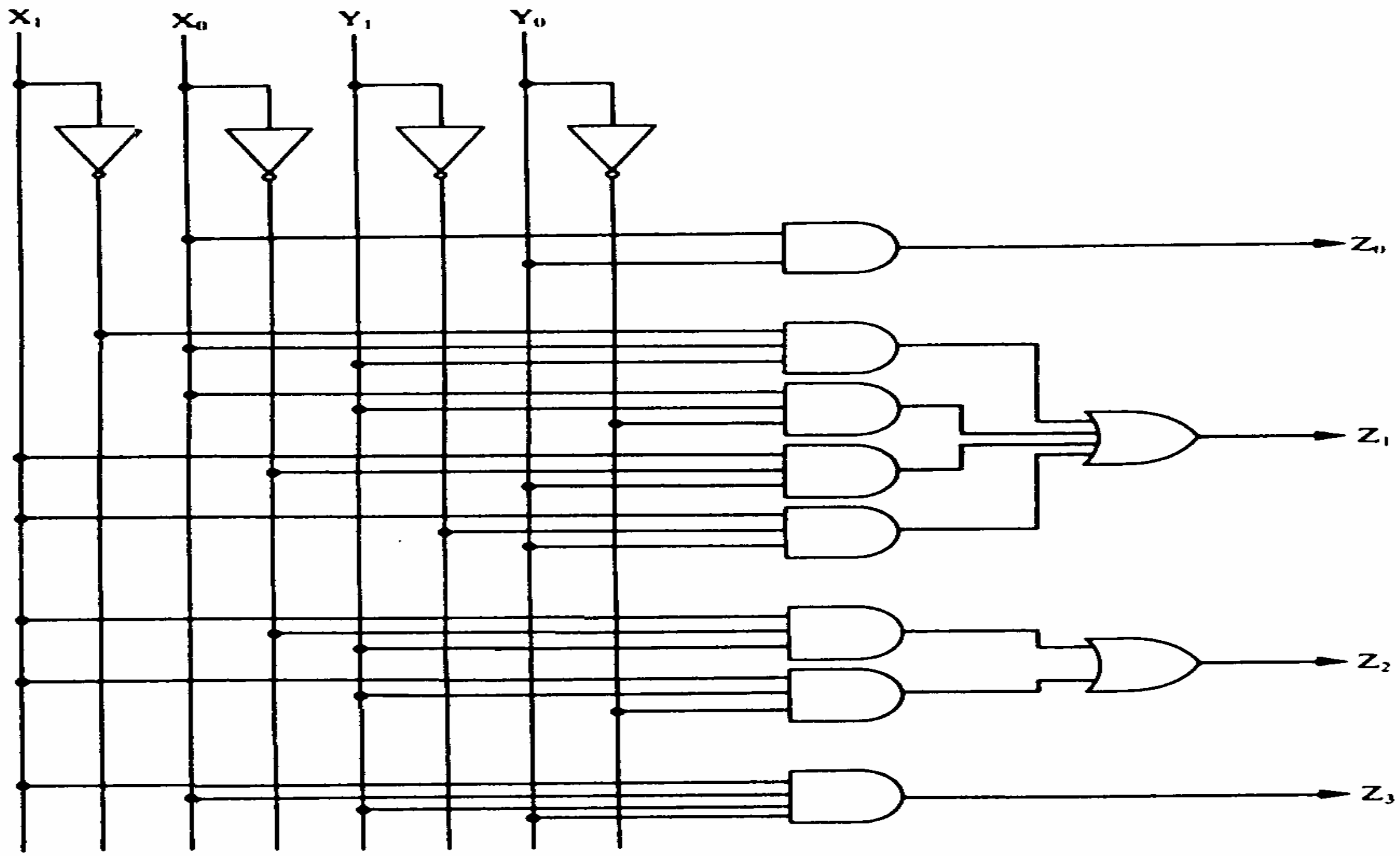



**Funktionstabelle:**

| | Inputs | | | | Outputs | | | |
|---|---|---|---|---|---|---|---|---|
| | X | | Y | | Z | | | |
| X×Y=Z | $X_1$ | $X_0$ | $Y_1$ | $Y_0$ | $Z_3$ | $Z_2$ | $Z_1$ | $Z_0$ |
| 0×0=0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0×1=0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0×2=0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0×3=0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1×0=0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1×1=1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1×2=2 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1×3=3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 2×0=0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2×1=2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 2×2=4 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 2×3=6 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 3×0=0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3×1=3 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 3×2=6 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 3×3=9 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

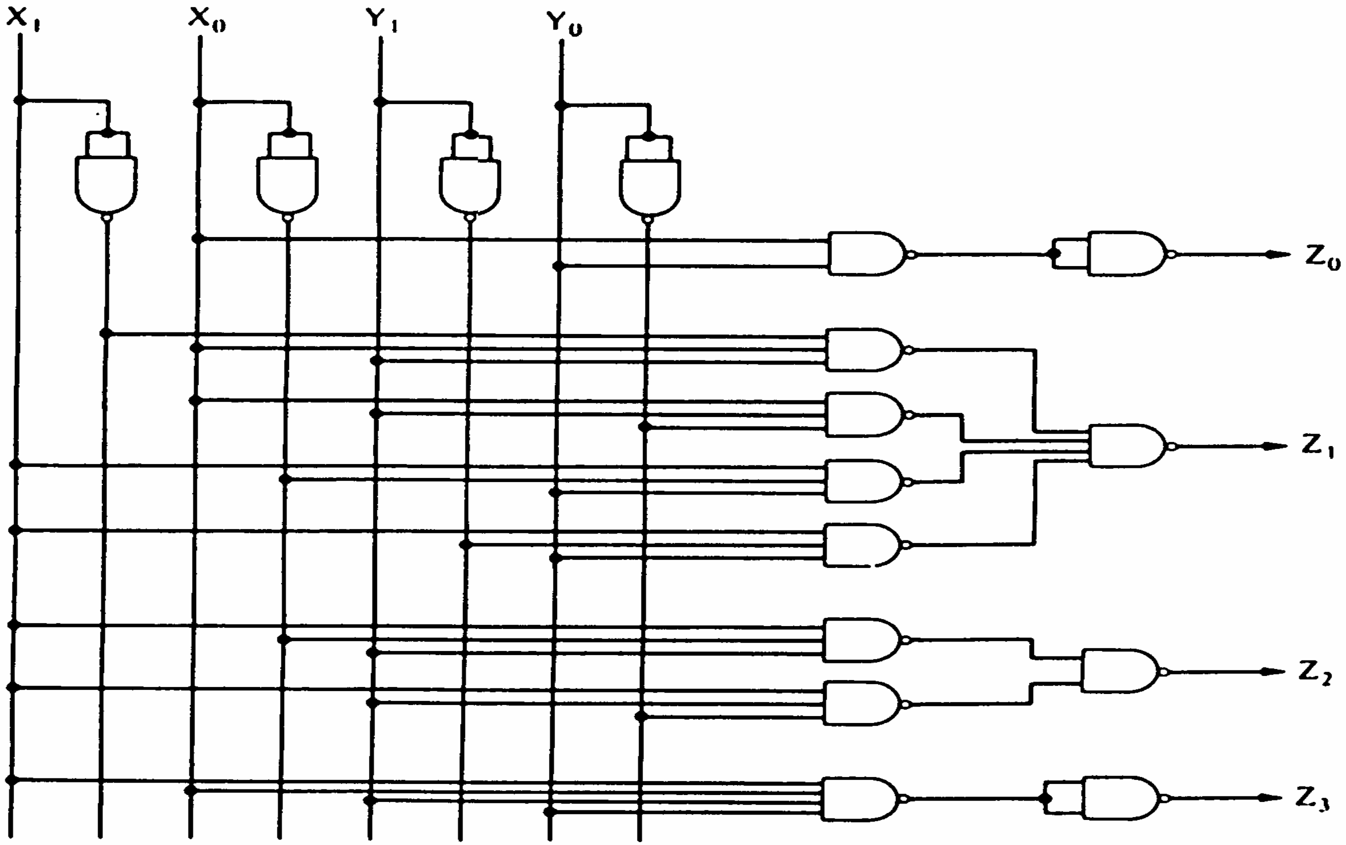
# Boolesche Algebra (4)

**Schaltnetz:**



# Boolesche Algebra (5)

**Äquivalentes Schaltnetz aus NAND-Gattern:**



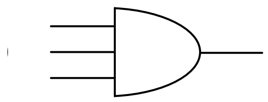
# Boolesche Algebra (6)

## Wichtig: NAND (NOR) - Operation ist nicht assoziativ!

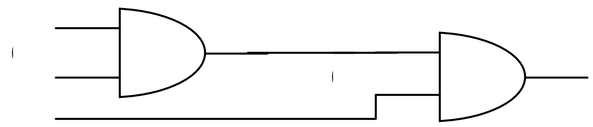
**--->** Logikentwurf mit NAND □(NOR) - Gattern nicht immer geradlinig, so z.B. bei der Realisierung von NAND-Funktionen mit n Variablen durch Standard m-input-NAND-Gatter (n>m)

**Beispiel:**

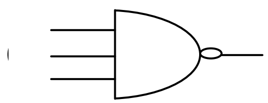
**a) Realisierung eines 3-input-AND mit 2-input-Gattern**



=

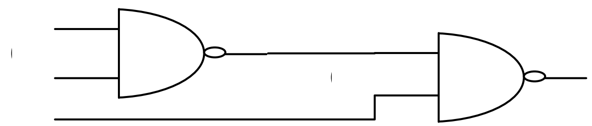


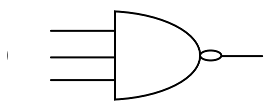
**b) Realisierung eines 3-input-NAND mit 2-input-Gattern**



=

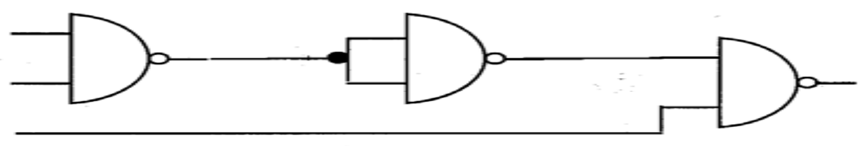
FALSCH





=

RICHTIG



# Boolesche (Schalt-) Algebra (7)

**Zusammenfassung**

- Schaltfunktionen sind ein logischer Formalismus (BA) zur Analyse und Synthese von Schaltnetzen und werden beschrieben mit Hilfe von Booleschen Ausdrücken
- Umformen von Booleschen Ausdrücken mit Hilfe der Axiome der BA dienen zur Analyse und Vereinfachung (Minimierung) von Schaltfunktionen (Schaltnetzen)
- Schaltnetze mit weniger Gattern und Inputs sind einfacher und kostengünstiger zu implementieren
- Es ist generell das Ziel, minimale Schaltnetze zu bestimmen

**Probleme bei der algebraischen Minimierung:**

- es gibt keine klare, methodische Vorgehensweise (Algorithmus), wie welche Regeln in welcher Reihenfolge angewandt werden sollen
- es ist nicht immer leicht zu bestimmen, wann ein Boolescher Ausdruck minimal ist

**mögliche Lösung:**

- Karnaugh-Diagramme
- Quine/McCluskey - Verfahren (bei mehr als 6 Variablen)