

# LOGIK

# Logik

## Gliederung

- Wozu Logik ?
- Boolesche Algebra
- Realisierung von der Logiksfunktion
- ICs Aufbauen

# Logik

- Wozu Logik ?
- Boolesche Algebra
- Realisierung von der Logiksfunktion
- ICs Aufbauen

# Wozu logik?

- „Wenn wir Umgangssprachlich sagen: ist doch logisch, dann heißt das: das muss so gelten, unabhängig vom sonstigen Kontext, von persönlichen Vorlieben oder Abneigungen, von herrschenden Meinungen oder Vorurteilen. Logik ist aus dieser Sicht die **Reflexion des Freien Denkens.**“ (Spies, Marcus 2004, Einleitung)

# Logik

- Wozu Logik ?
- **Boolesche Algebra**
- Realisierung von der Logiksfunktion
- ICs Aufbauen

# Boolesche Algebra

- Definition:

In der Mathematik ist eine **boolesche Algebra** (oder ein **boolescher Verband**) eine spezielle algebraische Struktur, die die Eigenschaften der logische Operatoren **UND**( $\wedge$ ) , **ODER**( $\vee$ ) , **NICHT**( $\neg$ ) verallgemeinert.

- Rechenregel von Boolesche Algebra

### Kommutativgesetze

$$a * b = b * a \quad \text{und} \quad a + b = b + a$$

### Assoziativgesetze

$$a * (b * c) = (a * b) * c$$

$$a + (b + c) = (a + b) + c$$

### Komplementärgesetz

$$a * \bar{a} = 0 \quad \text{und} \quad a + \bar{a} = 1$$

### De Morgansche Gesetze

$$\overline{a * b} = \bar{a} + \bar{b} \quad \text{und} \quad \overline{a + b} = \bar{a} * \bar{b}$$

### Doppelnegationsgesetz

$$\overline{\bar{a}} = a$$

### Extremalgesetze

$$a * 0 = 0 \quad \text{und} \quad a + 1 = 1$$

Wichtige Hinweis:

a , b hat nur zwei Zuständen. „0“ und „1“ .

also  $a \wedge b = a * b$  ,  $a \vee b = a + b$  .

- $Y = \overline{a} * b + b$



- $Y = \overline{a} * \overline{b} + b = \overline{a} + \overline{b} + b$

- $Y = \overline{a} * \overline{b} + b = \overline{a} + \overline{b} + b = \overline{a} + 1 = 1$

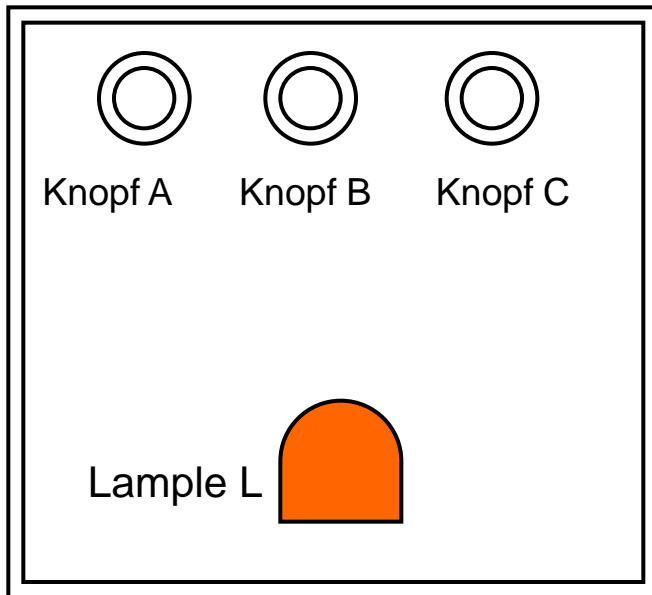
# Logik

- Wozu Logik ?
- Boolesche Algebra
- **Realisierung von der Logiksfunktion**
- ICs Aufbauen

# Wie kann man eine Logiksfunktion herleiten und realisieren?

- 1) Die Regelungsbedingungen
- 2) die Wahrheitstabelle
- 3)KV-Diagramm
- 4) Logiksfunktion wird erzeugt
- 5) Realisierung der Logikschaltungen

# Eine Beispiel:



- C gedrückt, L leuchtet. Aber nur wenn A und B beide nicht gedrückt.
- A und B beide gedrückt, C nicht gedrückt, L leuchtet auch.
- A, B und C gedrückt, L leuchtet nicht.
- Andere Fälle leuchtet L auch nicht.

# Wahrheitstabelle

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

# Wahrheitstabelle

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

# KV-Diagramm

	$\bar{C}$	C
$\bar{A}\bar{B}$		
$\bar{A}B$		
$A\bar{B}$		
AB		

# Wahrheitstabelle

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

# KV-Diagramm

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	
$\bar{A}B$		
$A\bar{B}$		
AB		



# Wahrheitstabelle

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

# KV-Diagramm

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$		
$A\bar{B}$		
AB		

# Wahrheitstabelle

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

# KV-Diagramm

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	0	1
$A\bar{B}$	0	1
AB	1	0

# Wahrheitstabelle

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

# KV-Diagramm

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	0	1
$A\bar{B}$	0	1
AB	1	0

# Wahrheitstabelle

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

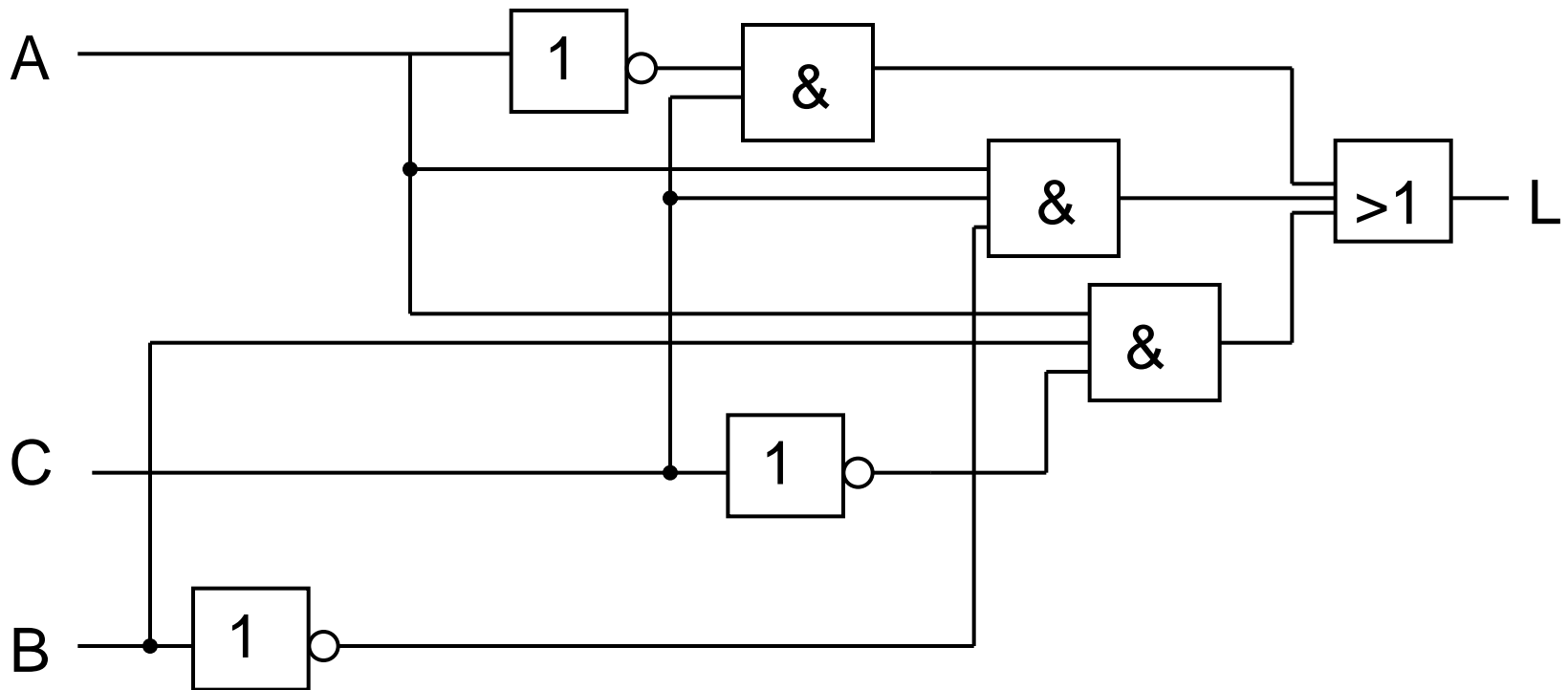
# KV-Diagramm

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	0	1
$A\bar{B}$	0	1
AB	1	0

$$L = \bar{A} * C + A * \bar{B} * C + A * B * \bar{C}$$

# Realisierung der Logikschaltung

$$L = \bar{A} * C + A * \bar{B} * C + A * B * \bar{C}$$



# Logik

- Wozu Logik ?
- Boolesche Algebra
- Wie kann man eine Logiksfunktion herleiten?
- **ICs Aufbauen**

# ICs Aufbauen

## Logikfamilien

- **RTL** – Resistor-Transistor-Logic
  - Widerständen und bipolaren Transistoren
- **DCTL** – Direct Coupled Transistor Logic
  - direkt miteinander gekoppelten bipolaren Transistoren
- **DRL** – Dioden-Resistor-Logic
  - Dioden und Widerständen
- **DTL** – Dioden-Transistor-Logic
  - Dioden und Transistoren
- **TTL** – Transistor-Transistor-Logic
  - Bipolaren Transistoren
- **MOS** – Metal Oxid Semiconductor
  - MOS-FETs

# Unterschiede zwischen TTL und CMOS

## TTL

- Bipolare Transistoren
- Betriebsspannung  
 $5V \pm 0.25V$
- Eingang  
L-Pegel:  $0V - 0.8V$   
H-Pegel:  $2V - 4.8V$
- Ausgang  
L-Pegel:  $< 0.4V$   
H-Pegel:  $> 2.4V$

## CMOS

- Selbstsperrende MOS-FETs
- Betriebsspannung festlegbar  
zwischen  $+3V$  bis  $+15V$
- Eingang  
L-Pegel:  $0 - 30\% U_B$   
H-Pegel:  $70\% - 100\% U_B$
- Ausgang  
L-Pegel:  $\sim 0V$   
H-Pegel:  $\sim U_B$



# Vor- und Nachteile

## TTL

- Ständiger Stromfluss
  - hohe Verlustleistung
  - Erwärmung
  - Begrenzte Komplexität
- 
- + Oft billiger
  - + Unempfindlichkeit
  - + Schnelligkeit

# Vor- und Nachteile

## CMOS

- Relativ große Schaltzeiten
- Empfindlich gegen statische Aufladung
- Empfindlich auf Eingangsspannungen über Versorgungsspannung oder unter 0V-Versorgungspotential
- Bei hohen Taktfrequenzen ( $> 5$  MHz): hohe Verlustleistung

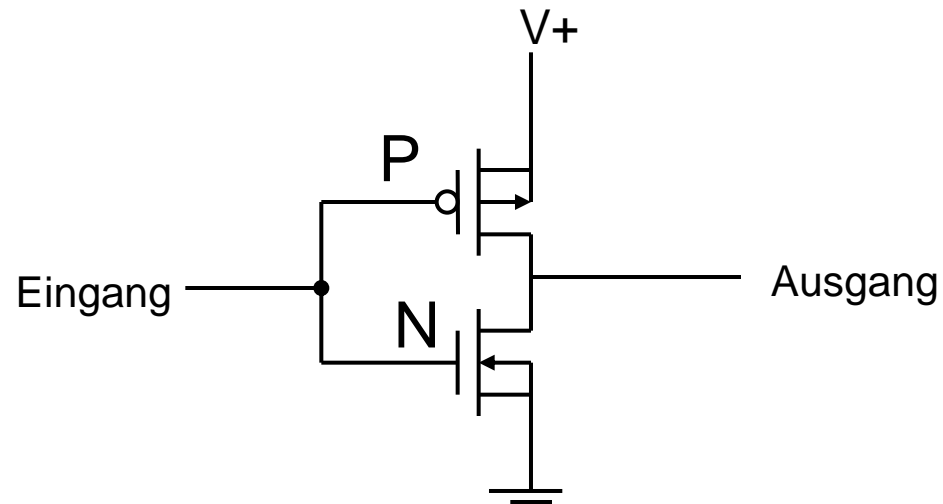
# Vor- und Nachteile

## CMOS

- + Keine Widerstände benötigt
- + Einfacherer Aufbau
- + Kein ständiger Stromfluss
- + Leistungsbedarf extrem niedrig ( $< 5 \text{ MHz}$ )
- + Festlegbare Betriebsspannung

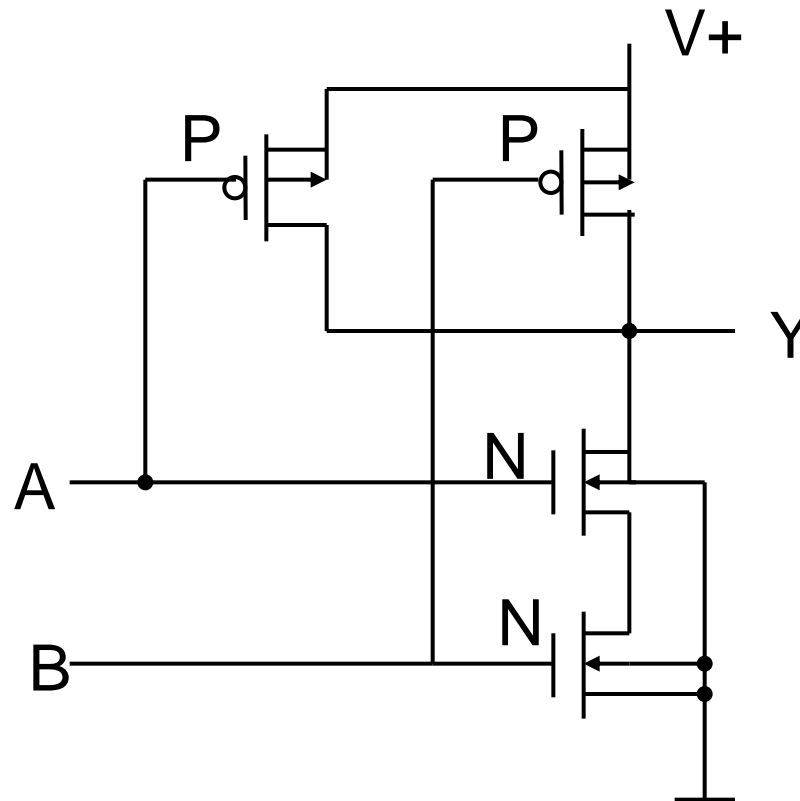
# Inverter( $A=\overline{E}$ )

- Mit CMOS



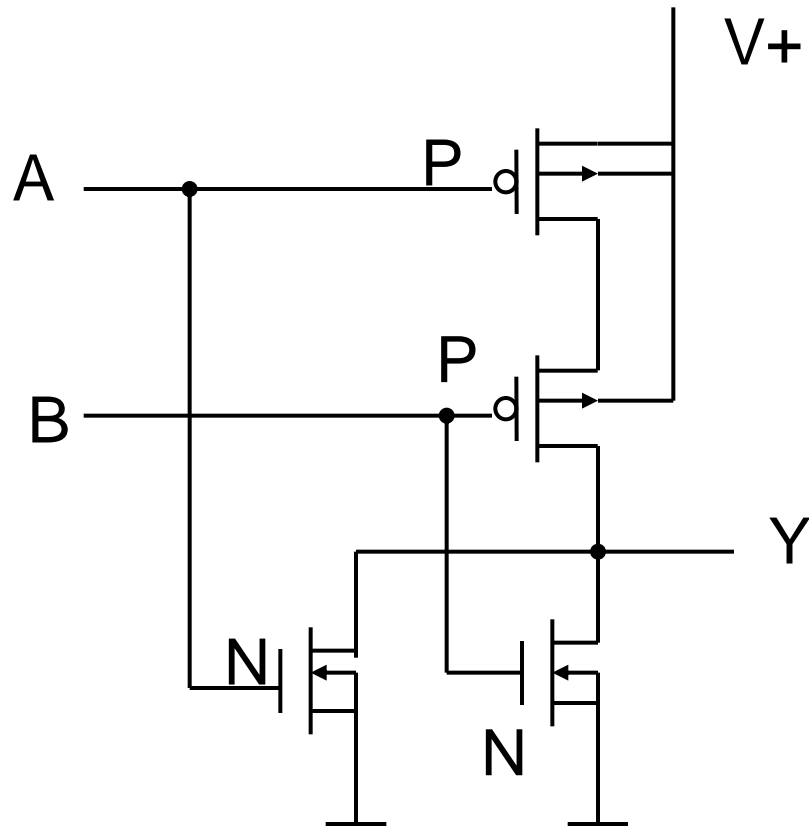
# NAND( $L=A*B$ )

Mit CMOS



# NOR( $L = \overline{A+B}$ )

Mit COMS



# Schaltungsaufbauten von eine

## Beispiele $L=A*B+C$

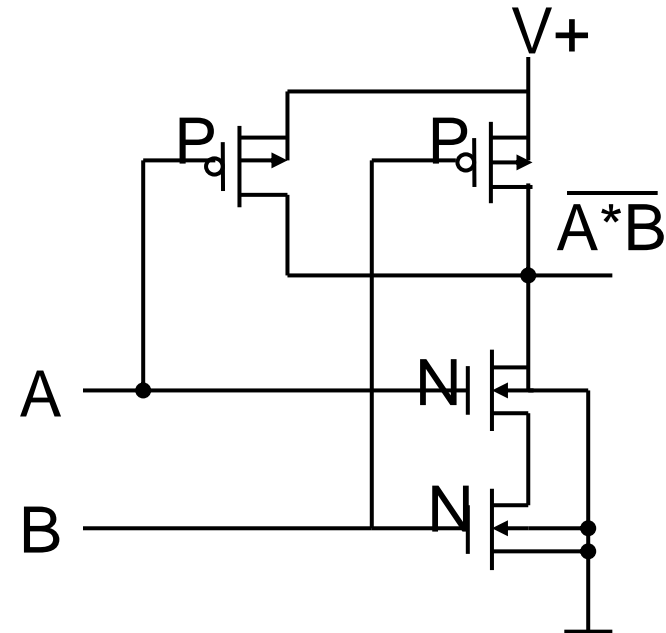
Mit Doppelnegationsgesetz kann man L so umformeln

$$\overline{\overline{A*B}} = A*B \quad , \quad L = \overline{\overline{A*B}} + C \quad , \quad L = \overline{\overline{\overline{\overline{A*B}}}} + C$$

# Realisierung von $L=A*B+C$

$$L=\overline{\overline{A*B} + C}$$

Mit CMOS realisiert

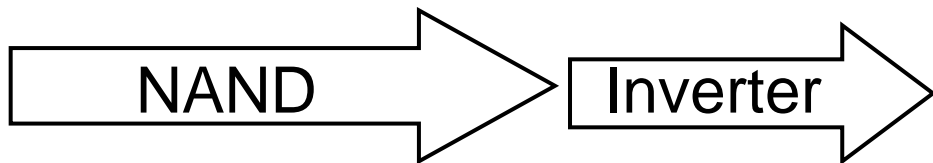
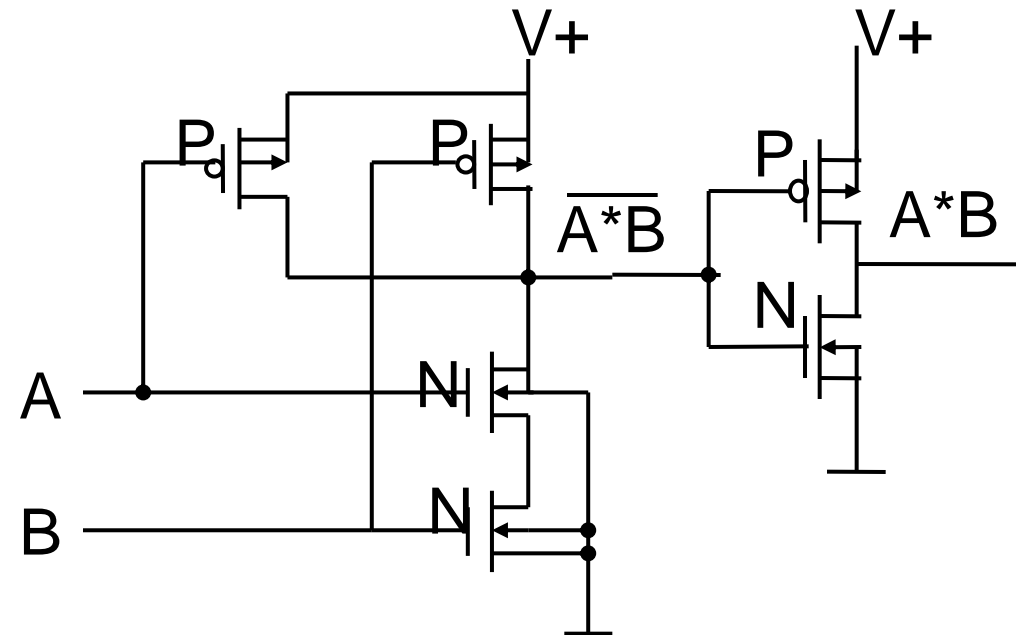




# Realisierung von $L=A*B+C$

$$L = \overline{\overline{A*B} + C}$$

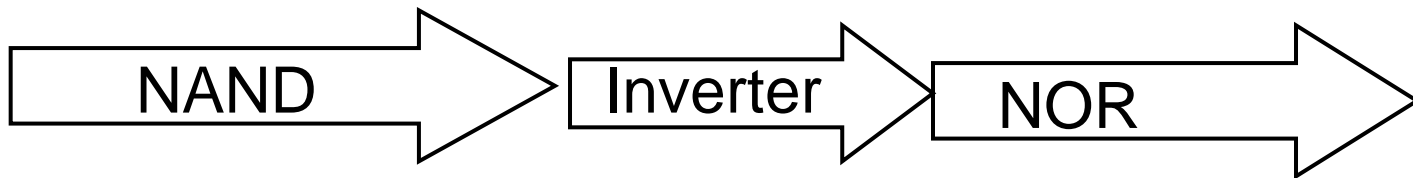
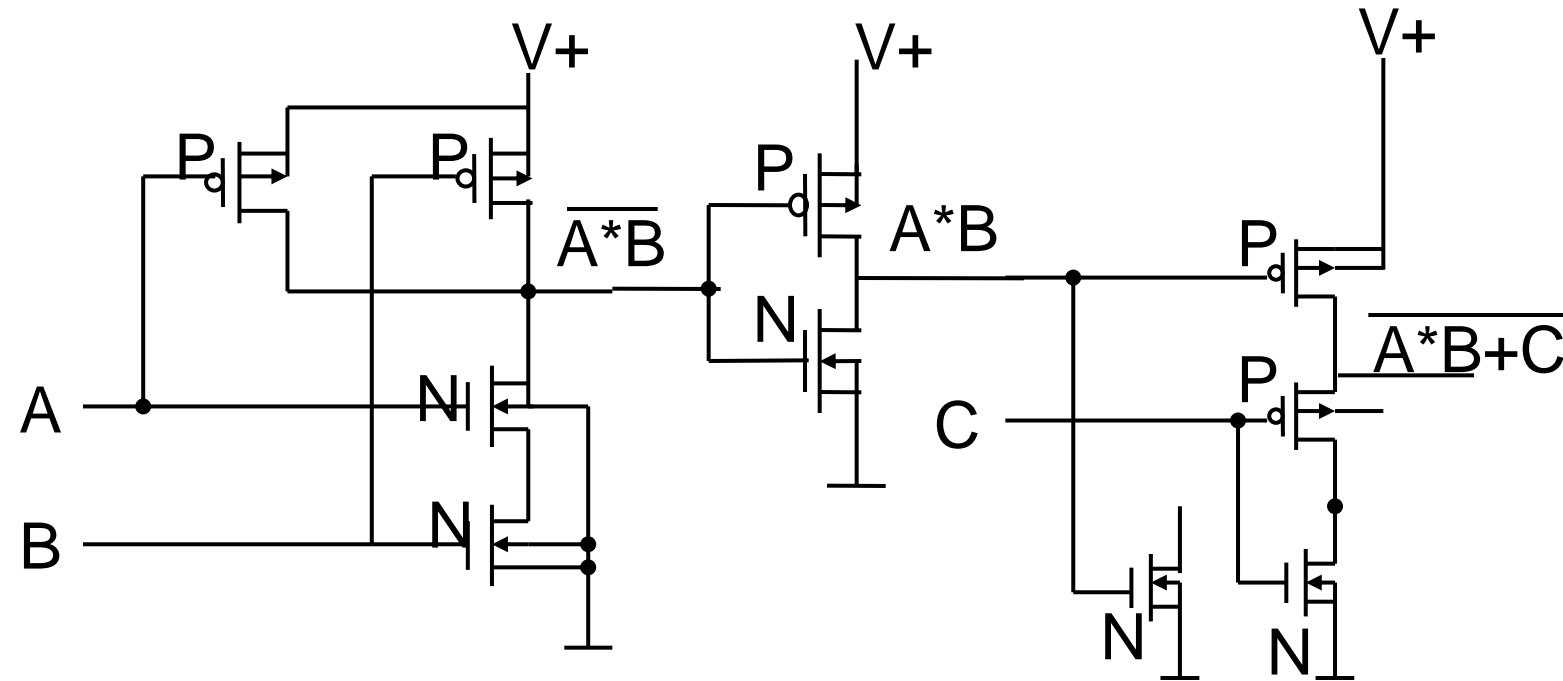
Mit CMOS realisiert



# Realisierung von $L=A*B+C$

$$L=\overline{\overline{A*B} + C}$$

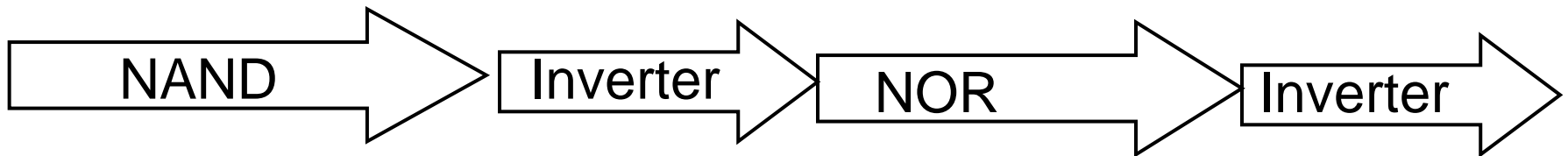
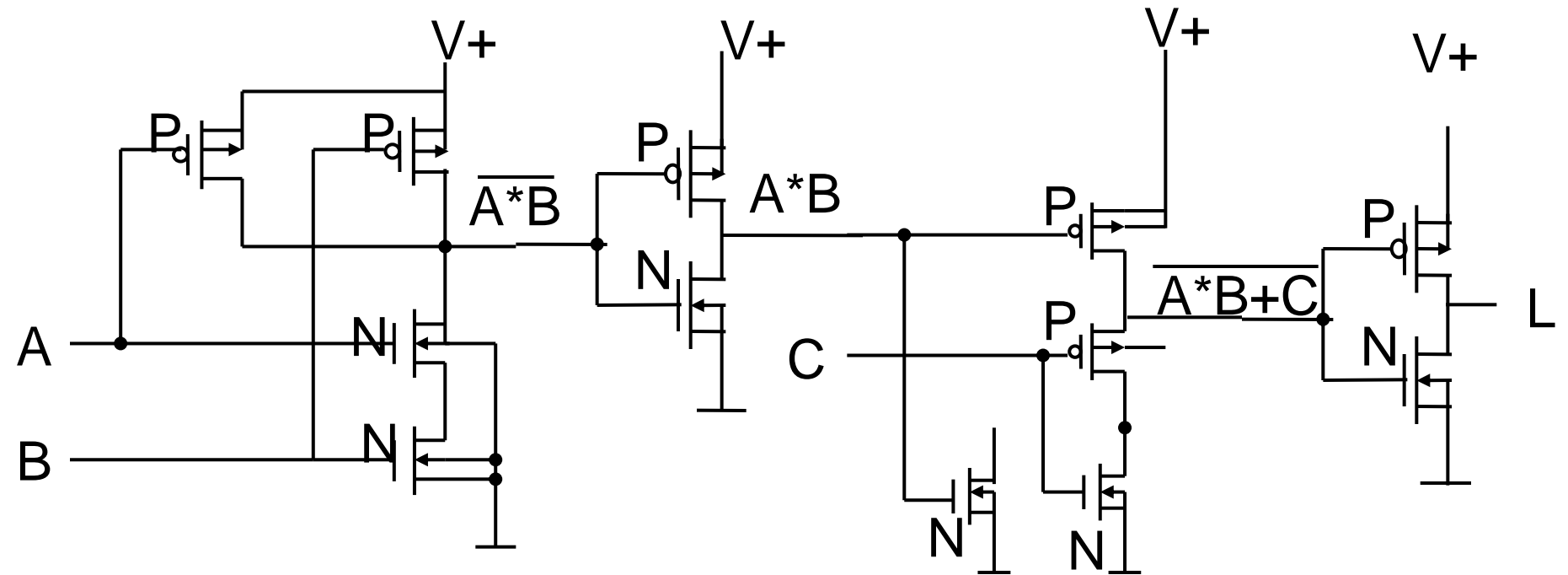
Mit CMOS realisiert



# Realisierung von $L=A*B+C$

$$L=\overline{\overline{A*B} + C}$$

Mit CMOS realisiert



- Quellen:

- [http://de.wikipedia.org/wiki/Boolesche\\_Algebra](http://de.wikipedia.org/wiki/Boolesche_Algebra)
- Orglmeister, Reinhold: Mikroprozessortechnik.
- Staab, frank : Logik und Algebra
- Spies, Marcus : Einführung in der Logik
- Glesner, Sabine : Einführung der Informatik I

# Dank für eure Aufmerksamkeit !

- Fragen??