Digitaltechnik
Wintersemester 2017/2018
5. Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

Inhalt

TECHNISCHE UNIVERSITÄT DARMSTADT

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Organisatorisches

TECHNISCHE UNIVERSITÄT DARMSTADT

- Anmeldung zu Studienleistung und Fachprüfung offen
- mündliche Prüfung für Austauschstudenten

Rückblick auf die letzten Vorlesungen

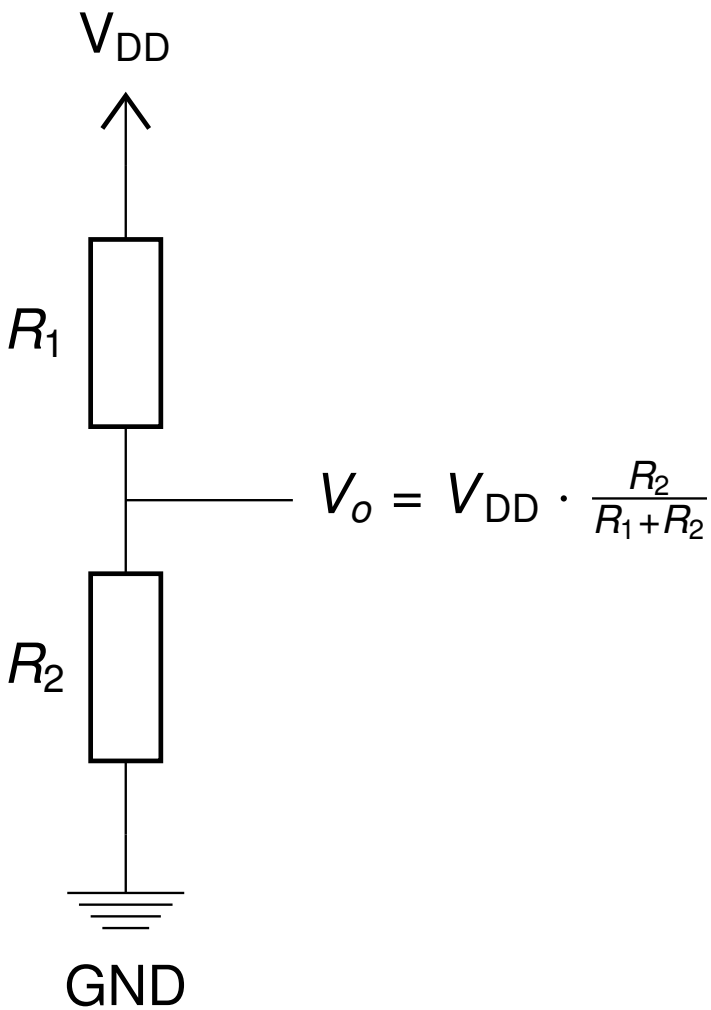
TECHNISCHE UNIVERSITÄT DARMSTADT

- Komplexität und (digitale) Abstraktion
- Zahlensysteme
- Logikgatter
- Physikalische Realisierung von Logikgattern
  - Logikpegel
  - Feldeffekt-Transistoren
  - CMOS-Gatter
  - Leistungsaufnahme
  - Moor'sches Gesetz
- Kombinatorische Logik
  - Boole'sche Gleichungen
  - Boole'sche Algebra

Harris 2013
Kapitel 1,2

Konzept des Spannungsteilers

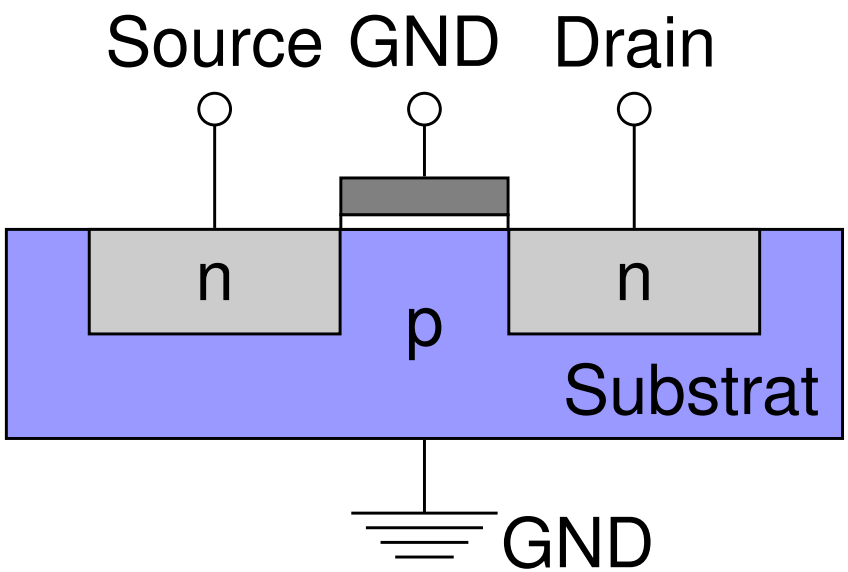
TECHNISCHE UNIVERSITÄT DARMSTADT

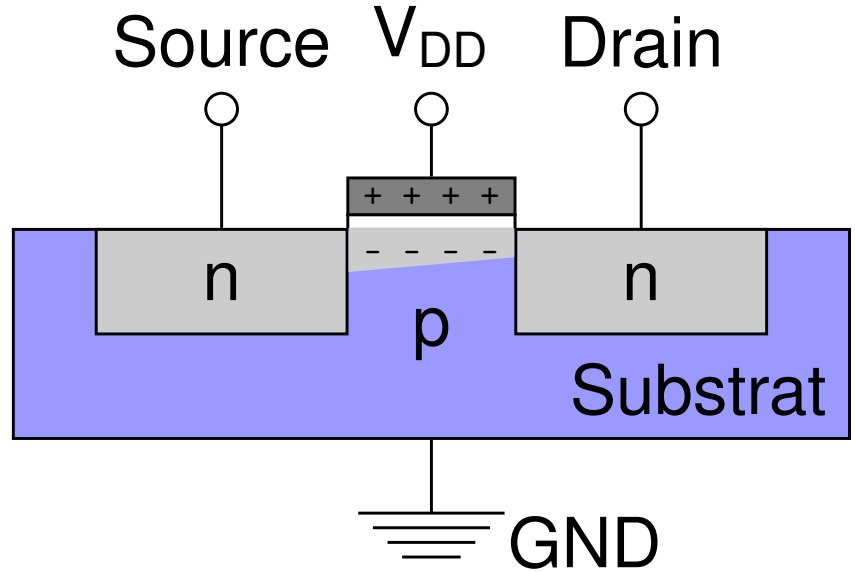


Wiederholung: nMOS

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gate = 0, ausgeschaltet (R_{sd} sehr groß)
- Gate = 1, eingeschaltet (R_{sd} sehr klein)

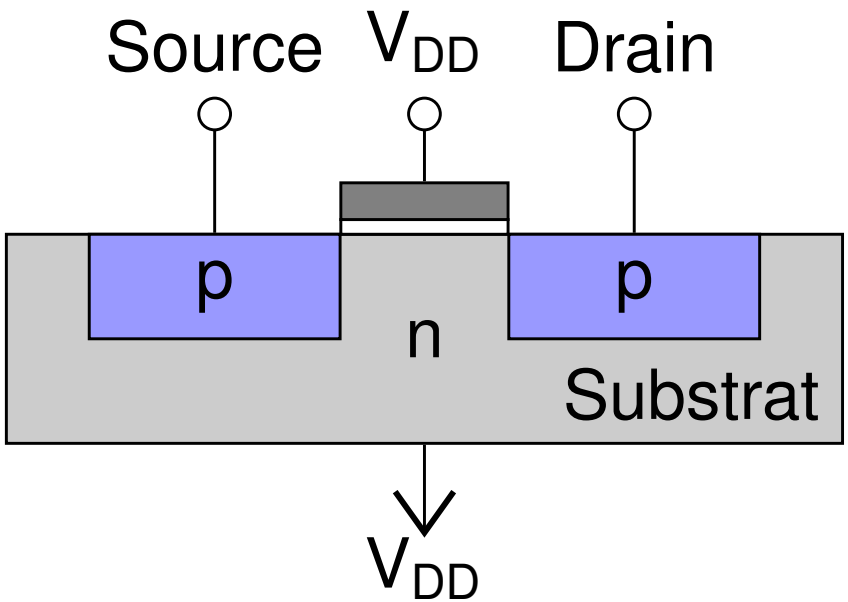


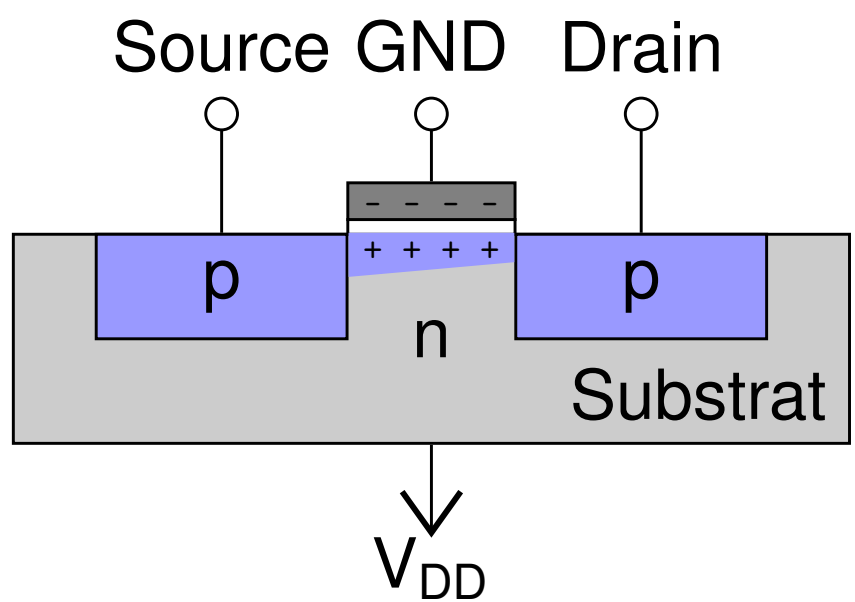


Wiederholung: pMOS

TECHNISCHE UNIVERSITÄT DARMSTADT

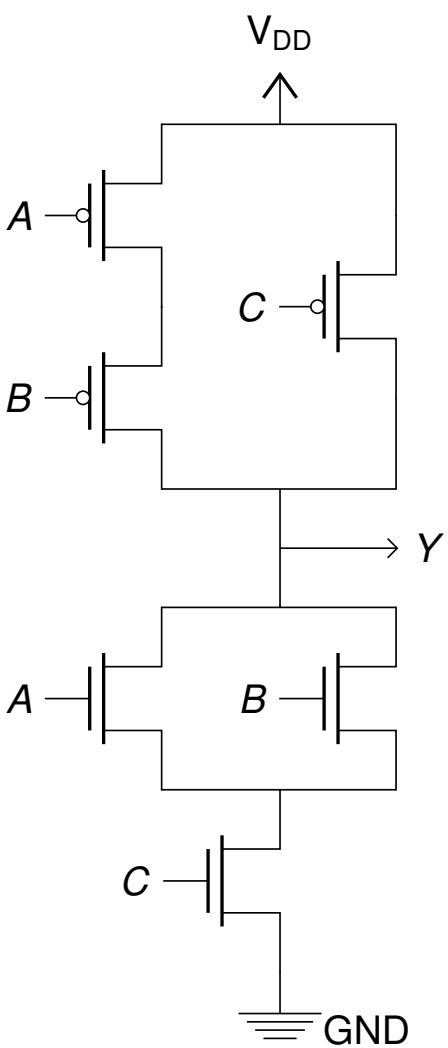
- Gate = 1, ausgeschaltet (R_{sd} sehr groß)
- Gate = 0, eingeschaltet (R_{sd} sehr klein)

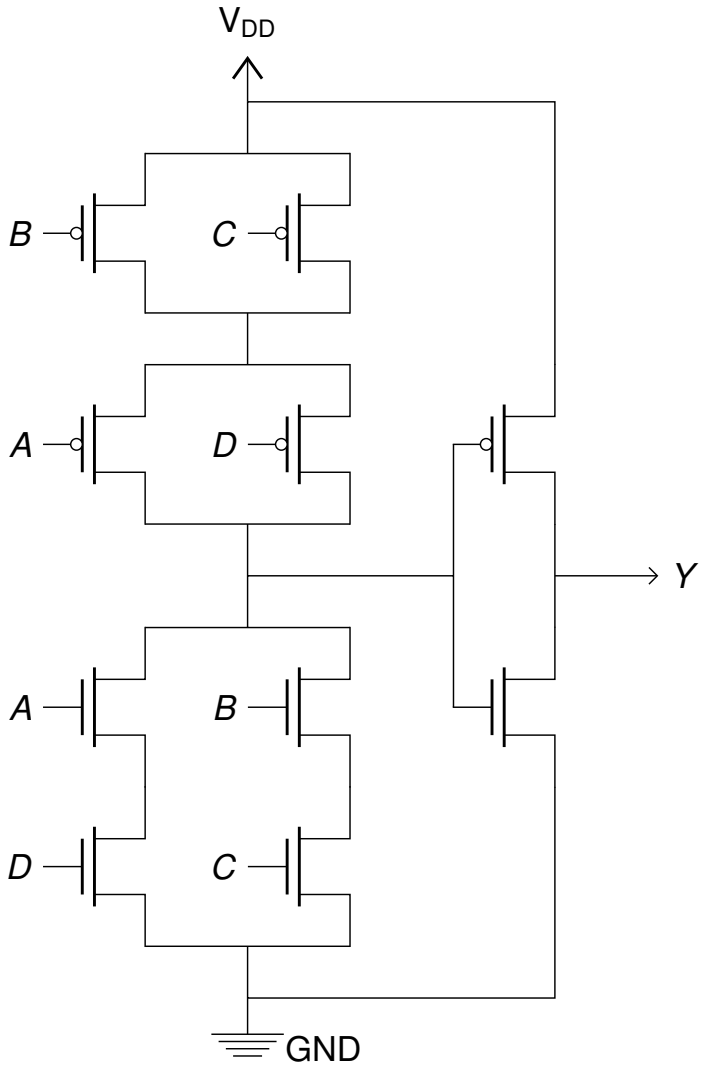


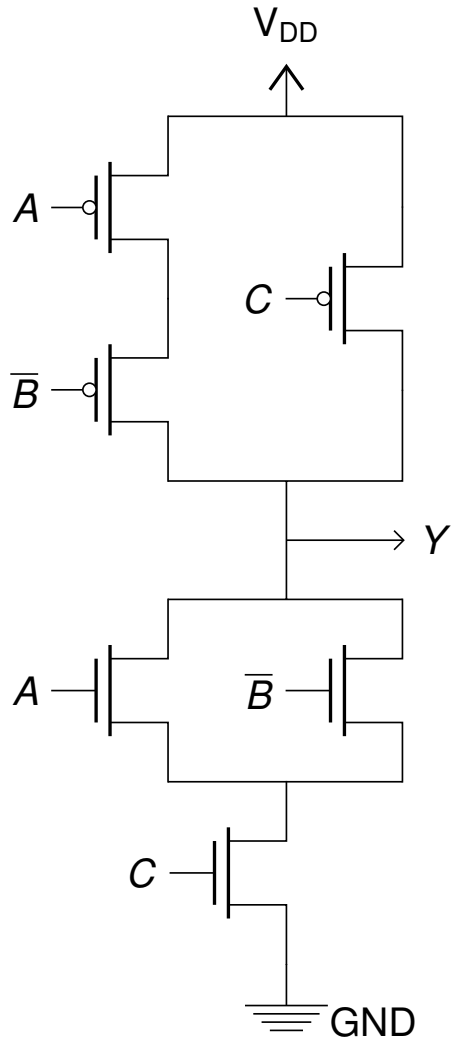


Wiederholung: CMOS

TECHNISCHE UNIVERSITÄT DARMSTADT







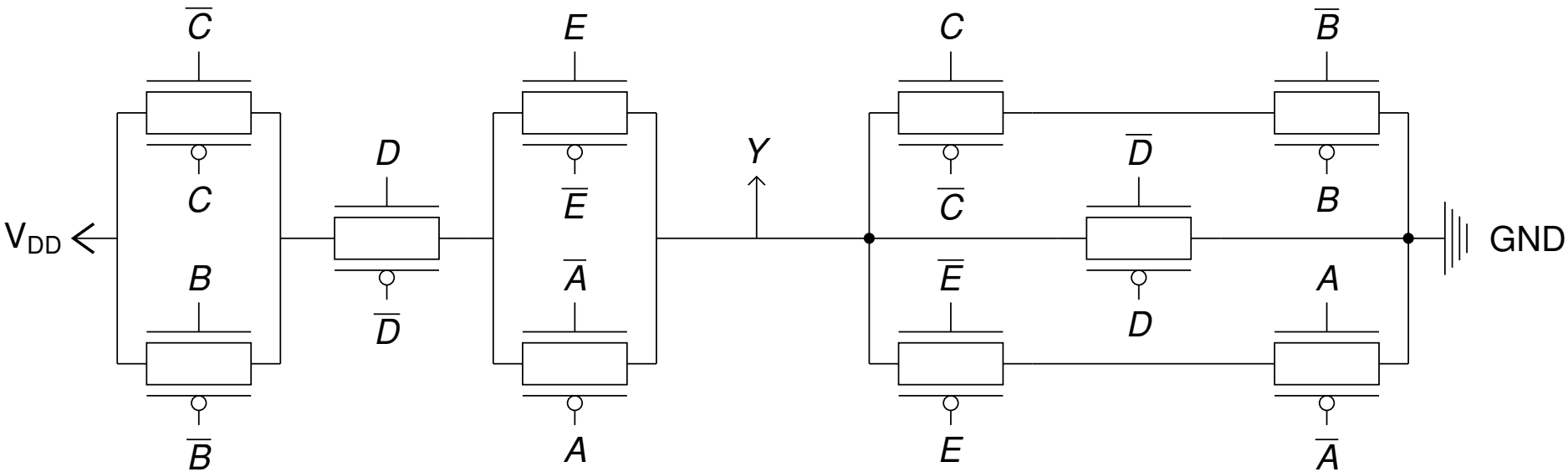
Konstruktion von CMOS-Schaltungen

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Y enthält nur negierten Variablen (bspw. $Y = \overline{A} + \overline{B}$)
   - pMOS konstruieren (OR parallel, AND seriell)
   - nMOS komplementär
   - keine negierten Variablen an Gates
2. $\overline{Y}$ enthält nur nicht-negierten Variablen (bspw. $\overline{Y} = A + B$)
   - nMOS konstruieren (OR parallel, AND seriell)
   - pMOS komplementär
   - keine negierten Variablen an Gates
3. Y enthält nur nicht-negierten Variablen (bspw. $Y = A + B$)
   - Konstruktion wie 2.
   - zusätzlicher Inverter am Ausgang
4. $\overline{Y}$ enthält nur negierten Variablen (bspw. $\overline{Y} = \overline{A} + \overline{B}$)
   - Konstruktion wie 1.
   - zusätzlicher Inverter am Ausgang
5. sonst (bspw. $Y = \overline{A} + B$)
   - negierte Literale an Gates verwenden
   - bspw. $Y = \overline{A} + \overline{\overline{B}}$

Wiederholung: Transmissionsgatter

TECHNISCHE UNIVERSITÄT DARMSTADT



Theoreme der boole'schen Algebra

TECHNISCHE UNIVERSITÄT DARMSTADT

	Theorem		Duales Theorem	Bedeutung
T1	$A \cdot 1 = A$	T1'	$A + 0 = A$	Neutralität
T2	$A \cdot 0 = 0$	T2'	$A + 1 = 1$	Extremum
T3	$A \cdot A = A$	T3'	$A + A = A$	Idempotenz
T4	$\overline{\overline{A}} = A$			Involution
T5	$A \cdot \overline{A} = 0$	T5'	$A + \overline{A} = 1$	Komplement
T6	$A \cdot B = B \cdot A$	T6'	$A + B = B + A$	Kommutativität
T7	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	T7'	$A + (B + C) = (A + B) + C$	Assoziativität
T8	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	T8'	$A + (B \cdot C) = (A + B) \cdot (A + C)$	Distributivität
T9	$A \cdot (A + B) = A$	T9'	$A + (A \cdot B) = A$	Absorption
T10	$(A \cdot B) + (A \cdot \overline{B}) = A$	T10'	$(A + B) \cdot (A + \overline{B}) = A$	Zusammenfassen
T11	$(A \cdot B) + (\overline{A} \cdot C) + (B \cdot C) = (A \cdot B) + (\overline{A} \cdot C)$	T11'	$(A + B) \cdot (\overline{A} + C) \cdot (B + C) = (A + B) \cdot (\overline{A} + C)$	Konsensus
T12	$\overline{A \cdot B \cdot C \ldots} = \overline{A} + \overline{B} + \overline{C} \ldots$	T12'	$\overline{A + B + C \ldots} = \overline{A} \cdot \overline{B} \cdot \overline{C} \ldots$	De Morgan

Beweis für Konsensus (T11) durch Anwendung von Axiomen und Theoremen

TECHNISCHE UNIVERSITÄT DARMSTADT

$$
\begin{aligned}
& A \cdot B + \overline{A} \cdot C + B \cdot C && \text{Neutralität} \\
= {} & A \cdot B + \overline{A} \cdot C + 1 \cdot B \cdot C && \text{Komplement} \\
= {} & A \cdot B + \overline{A} \cdot C + (A + \overline{A}) \cdot B \cdot C && \text{Distributivität} \\
= {} & A \cdot B + \overline{A} \cdot C + A \cdot B \cdot C + \overline{A} \cdot B \cdot C && \text{Kommutativität} \\
= {} & A \cdot B + A \cdot B \cdot C + \overline{A} \cdot C + \overline{A} \cdot C \cdot B && \text{Neutralität} \\
= {} & A \cdot B \cdot 1 + A \cdot B \cdot C + \overline{A} \cdot C \cdot 1 + \overline{A} \cdot C \cdot B && \text{Distributivität} \\
= {} & A \cdot B \cdot (1 + C) + \overline{A} \cdot C \cdot (1 + B) && \text{Extremum} \\
= {} & A \cdot B \cdot 1 + \overline{A} \cdot C \cdot 1 && \text{Neutralität} \\
= {} & A \cdot B + \overline{A} \cdot C && \text{q.e.d.}
\end{aligned}
$$

Überblick der heutigen Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Kombinatorische Logik
  - Bubble Pushing
  - Logik-Realisierung mit Basis-Gattern
  - Karnaugh Diagramme

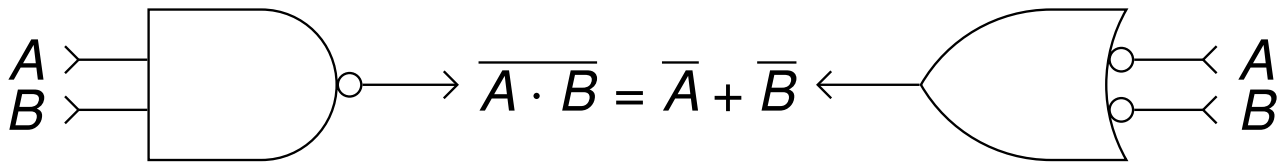
Harris 2013
Kap. 2.4,2.5,2.7,2.8

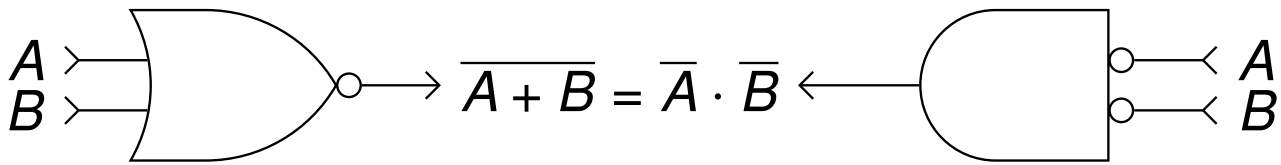
Bubble Pushing

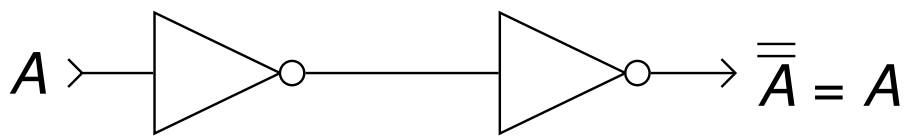
TECHNISCHE UNIVERSITÄT DARMSTADT

Graphische Umformung von Schaltungen nach De Morgan und Inversion

TECHNISCHE UNIVERSITÄT DARMSTADT







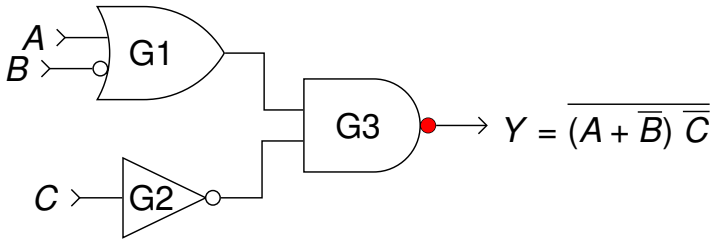
Invertierungsblasen verschieben
Bubble Pushing

TECHNISCHE UNIVERSITÄT DARMSTADT

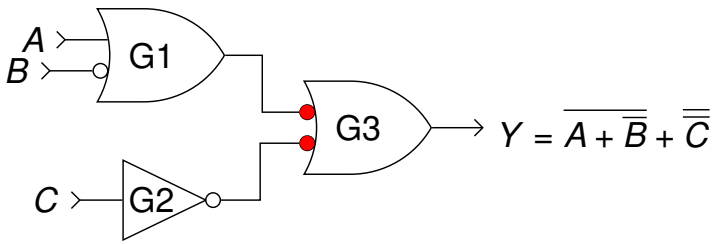
- über Gatter (AND/OR/NOT/BUF) hinweg
  - vorwärts: Eingang $\rightarrow$ Ausgang
  - rückwärts: Ausgang $\rightarrow$ Eingang
  - Art des Gatters ändern: AND $\leftrightarrow$ OR
  - Blasen an *allen* Eingängen ändern: vorhanden $\leftrightarrow$ nicht vorhanden
  - Blase an Ausgang ändert: vorhanden $\leftrightarrow$ nicht vorhanden
- zwischen Gattern
  - vorwärts: Treiber $\rightarrow$ *alle* Empfänger
  - rückwärts: *alle* Empfänger $\rightarrow$ Treiber
  - doppelte Blasen heben sich gegenseitig auf (Involution)
- verbleibende Buffer (vorher Inverter) können entfernt werden

Beispiel: Invertierungsblasen rückwärts verschieben

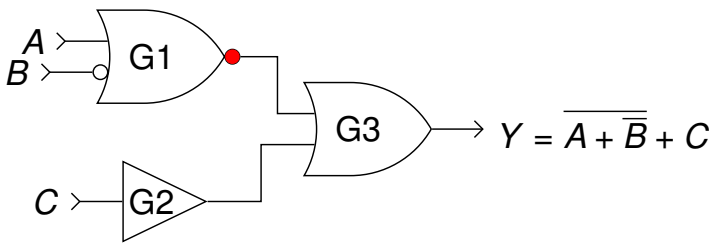
TECHNISCHE UNIVERSITÄT DARMSTADT



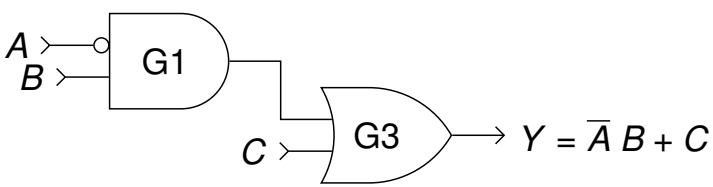
- De Morgan über G3
  - Blase am Ausgang $\rightarrow$ Blase an beiden Eingängen
  - AND $\rightarrow$ OR



- Blasen entlang Leitungen verschieben
  - G3 $\rightarrow$ G1
  - G3 $\rightarrow$ G2 (Doppelblase aufheben)



- De Morgan über G1
  - Blasen an Ein- und Ausgängen *invertieren*
  - OR $\rightarrow$ AND
- Buffer G2 entfernen



- zwei Inverter weniger

Wozu das Ganze?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Schaltungen vereinfachen
  - weniger Invertierer
  - weniger Literale
  - weniger verschiedene Gatter-Arten $\rightarrow$ einfachere Zellbibliothek
- Komplementäre Schaltungen ableiten
  - bspw. für CMOS- oder Transmissionsgatter-Schaltung

Logik-Realisierung mit Basis-Gattern

TECHNISCHE UNIVERSITÄT DARMSTADT

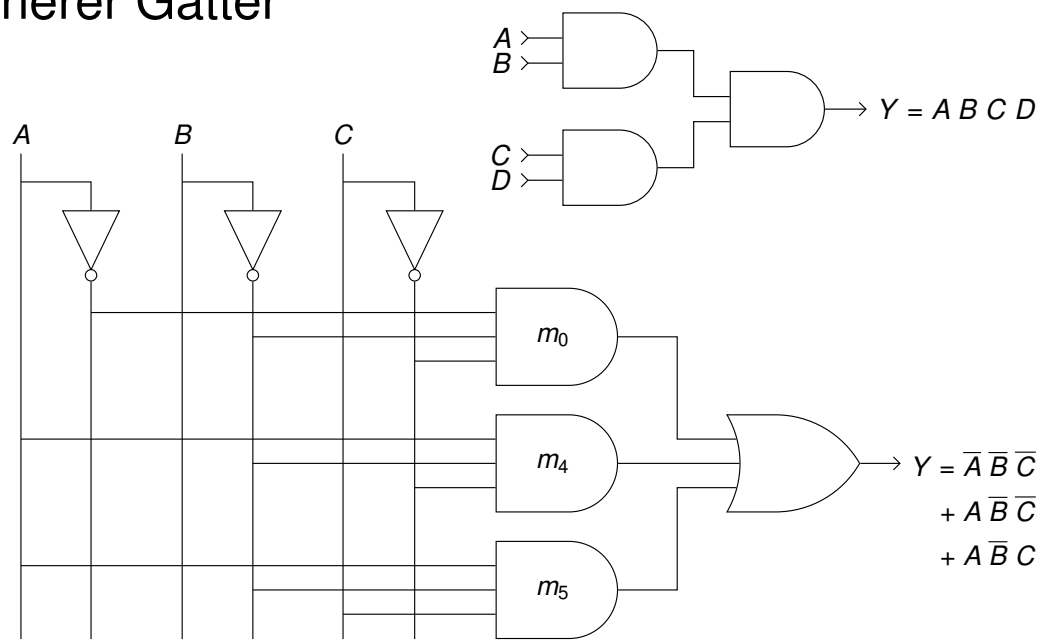
Zweistufige Logik

TECHNISCHE UNIVERSITÄT DARMSTADT

- direkte (konstruktive) Umsetzung der disjunktiven Normalform
  - Eingangsliterale: ein Inverter pro Variable
  - Minterme: je ein „breites“ AND Gatter an passende Literale anschließen
  - Summe: alle Minterme an ein „breites“ OR Gatter anschließen
- „breite“ Gatter als Kaskaden kleinerer Gatter

$\Rightarrow$ jede boole’sche Funktion realisierbar mit Basisgattern

- AND2
- OR2
- NOT



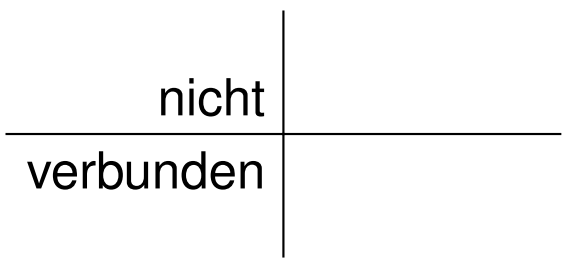
Konventionen für lesbare Schaltpläne

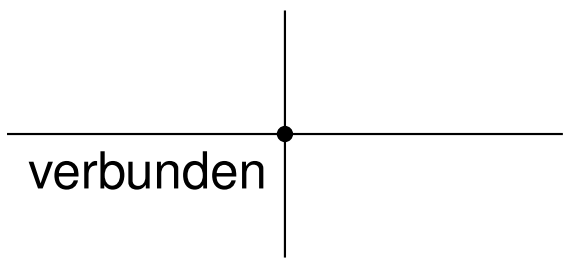
TECHNISCHE UNIVERSITÄT DARMSTADT

- Eingänge links (oder oberen)
- Ausgänge rechts (oder unteren)
- Gatter von links nach rechts (oben nach unten) angeordnet
- gerade (oder rechtwinklige) Verbundungen

$\Rightarrow$ keine Schrägen oder Kurven

- 3-armige Kreuzungen gelten implizit als verbunden
- 4-armige Kreuzungen gelten nur bei Markierung (Punkt) als verbunden





Weitere kombinatorische Grundelemente

TECHNISCHE UNIVERSITÄT DARMSTADT

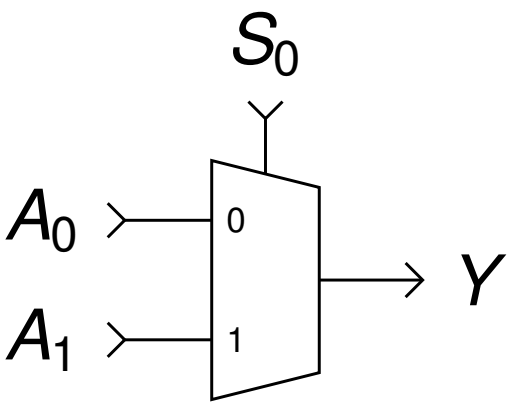
- zweistufige Logik
  - sehr mächtig
  - aufwändige Darstellung und Realisierung
  - realisiertes Verhalten nicht intuitiv ersichtlich
- weitere Basisgatter neben AND, OR, NOT:
  - XOR: Parität
  - Multiplexer: n zu 1 Auswahl
  - Dekodierer: n zu 2^n Auswahl („One-Hot“)

Multiplexer

MUX$n : \mathbb{B}^{n+\lceil \log_2 n \rceil} \rightarrow \mathbb{B}$

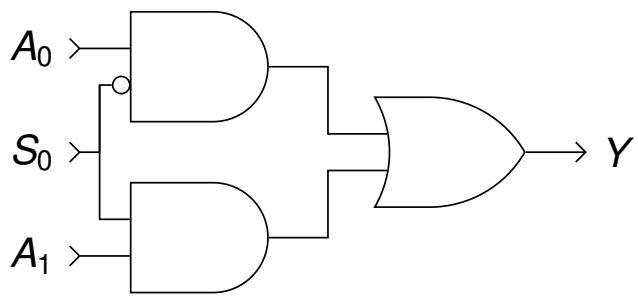
TECHNISCHE UNIVERSITÄT DARMSTADT

- verbindet einen von n Dateneingängen $A_0, \ldots, A_{n-1}$ mit Ausgang Y
- $k = \lceil \log_2 n \rceil$ Steuersignale $S_0, \ldots, S_{k-1}$
- $Y = A_{\mathrm{u}_{2,k}(S_{k-1} \ldots S_0)}$



S_0	A_0	A_1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

S_0	Y
0	A_0
1	A_1

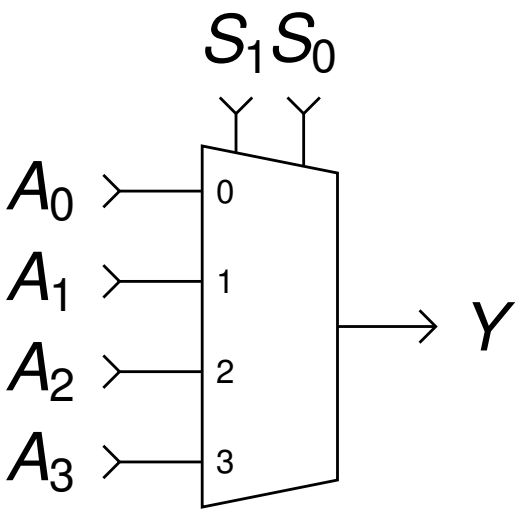


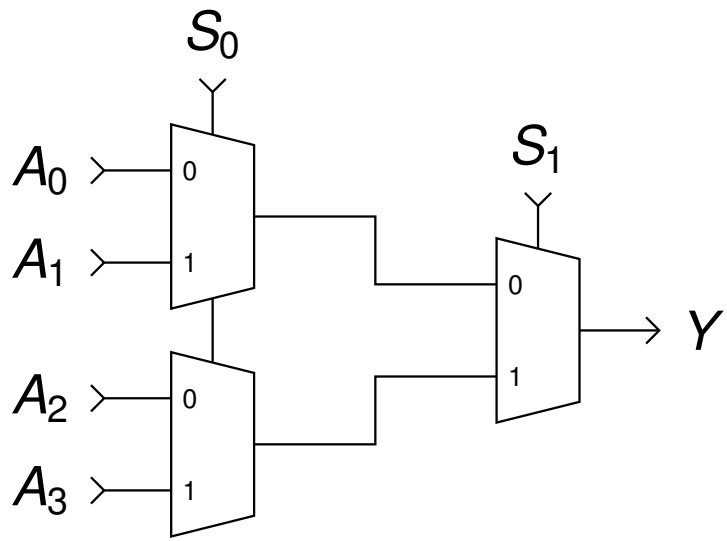
Multiplexer

MUX4 : $\mathbb{B}^6 \rightarrow \mathbb{B}$

TECHNISCHE UNIVERSITÄT DARMSTADT

S_1	S_0	Y
0	0	A_0
0	1	A_1
1	0	A_2
1	1	A_3



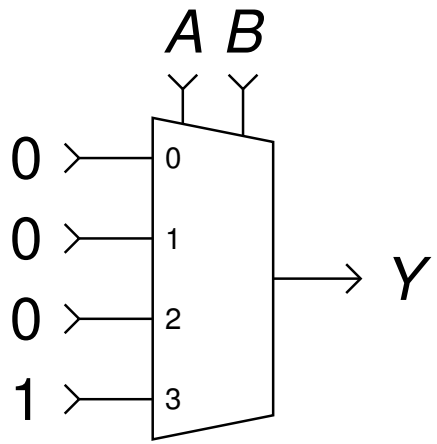


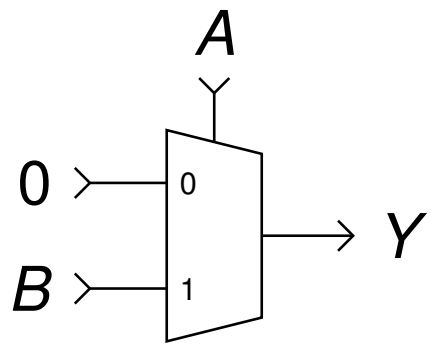
Logikrealisierung mit Multiplexern

TECHNISCHE UNIVERSITÄT DARMSTADT

- Variablen als Steuersignale verwenden
- Wahrheitswertetabelle als Konstanten an Dateneingängen
- entspricht adressiertem Speicherzugriff
  - Look-up Tabelle
  - ROM oder RAM $\rightarrow$ rekonfigurierbare Logik
- weitere funktionsspezifische Optimierungen möglich

A	B	$Y = A\,B$
0	0	0
0	1	0
1	0	0
1	1	1





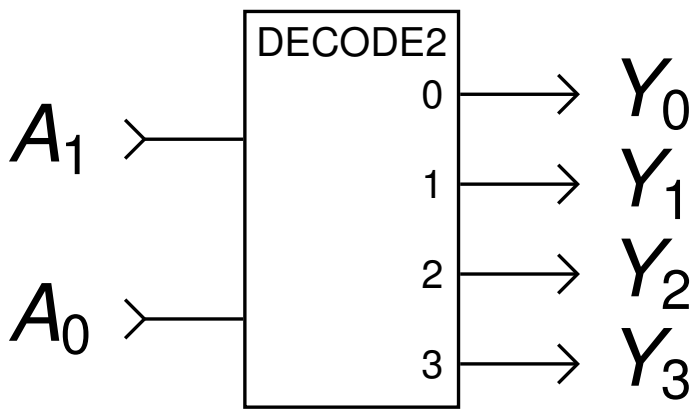
Dekodierer

DECODE$n : \mathbb{B}^n \rightarrow \mathbb{B}^{2^n}$

TECHNISCHE UNIVERSITÄT DARMSTADT

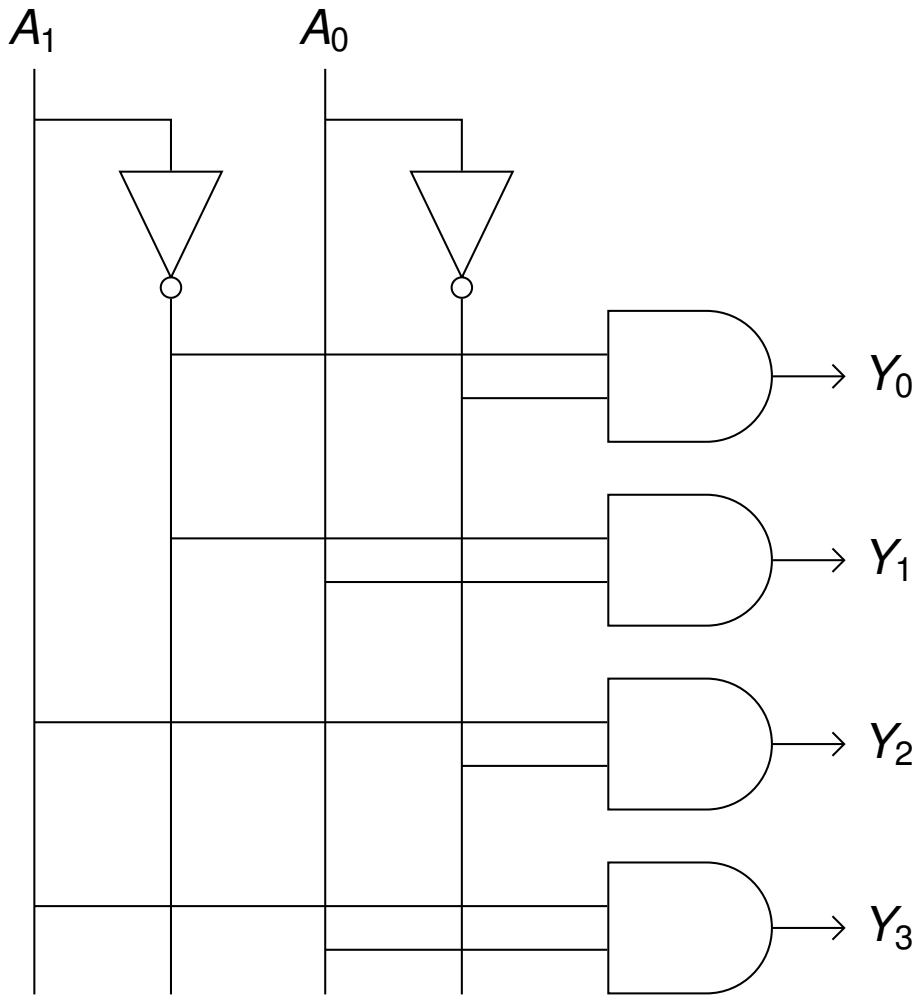
- n Eingänge $A_0, \ldots, A_{n-1}$
- 2^n Ausgänge $Y_0, \ldots, Y_{2^n-1}$
- „One-Hot“ Kodierung: $Y_i = \mathrm{u}_{2,n}(A_{n-1} \ldots A_0) == i \; ? \; 1 : 0$

A_1	A_0	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Implementierung von Dekodierern

TECHNISCHE UNIVERSITÄT DARMSTADT



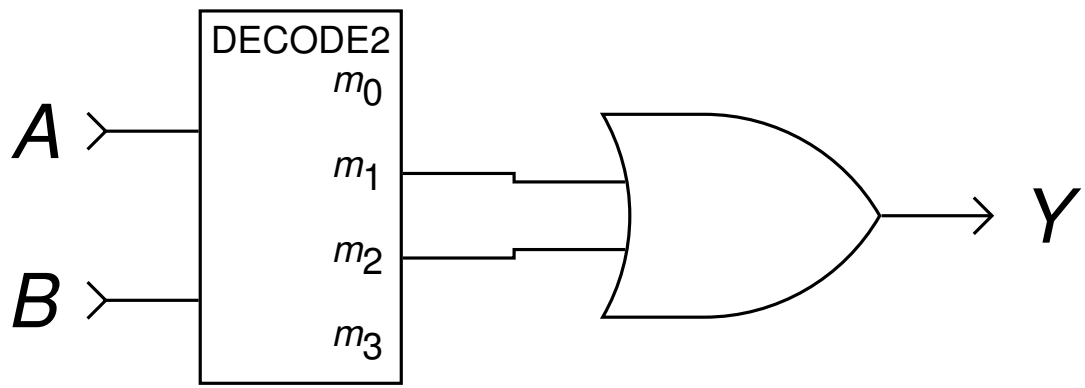
Logikrealisierung mit Decodern

TECHNISCHE UNIVERSITÄT DARMSTADT

- Summe über Minterme, auf denen Zielfunktion wahr ist

$\Rightarrow$ Decoder ersetzt erste Stufe der zweistufigen Logikrealisierung

A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



Karnaugh Diagramme

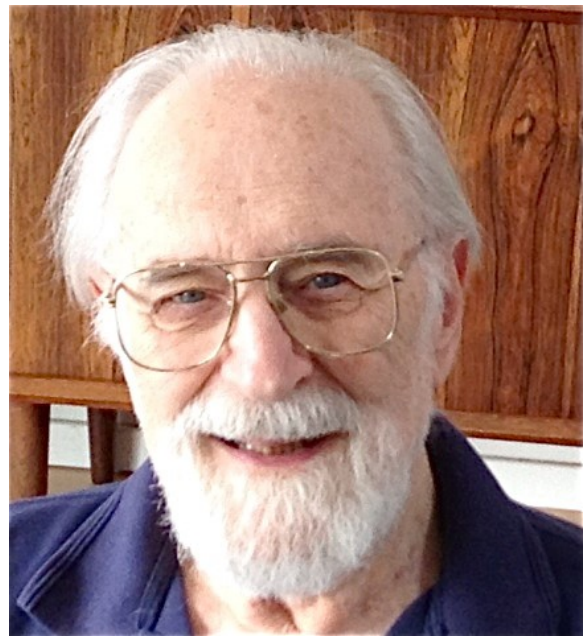
TECHNISCHE UNIVERSITÄT DARMSTADT

Maurice Karnaugh, 1924 -

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bell Laboratory
- IBM Research
- Techniken und Methoden für den schnellen Entwurf informationstechnischer Systeme

$\Rightarrow$ Karnaugh(-Veitch) Diagramme



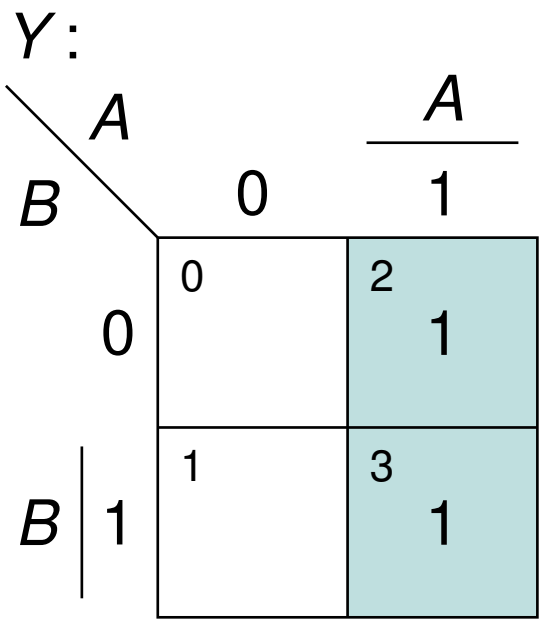
Karnaugh Diagramme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Boole'sche Ausdrücke können durch Zusammenfassen von Mintermen minimiert werden
  - $Y = A\,B + A\,\overline{B} = A$
- Karnaugh-Diagramme stellen Zusammenhänge graphisch dar
  - geschickte Anordnung der Wahrheitswertetabelle
  - benachbarte Einträge gehören zu gleichem Literal

  $\Rightarrow$ Zusammenhängende Minterme besser erkennbar

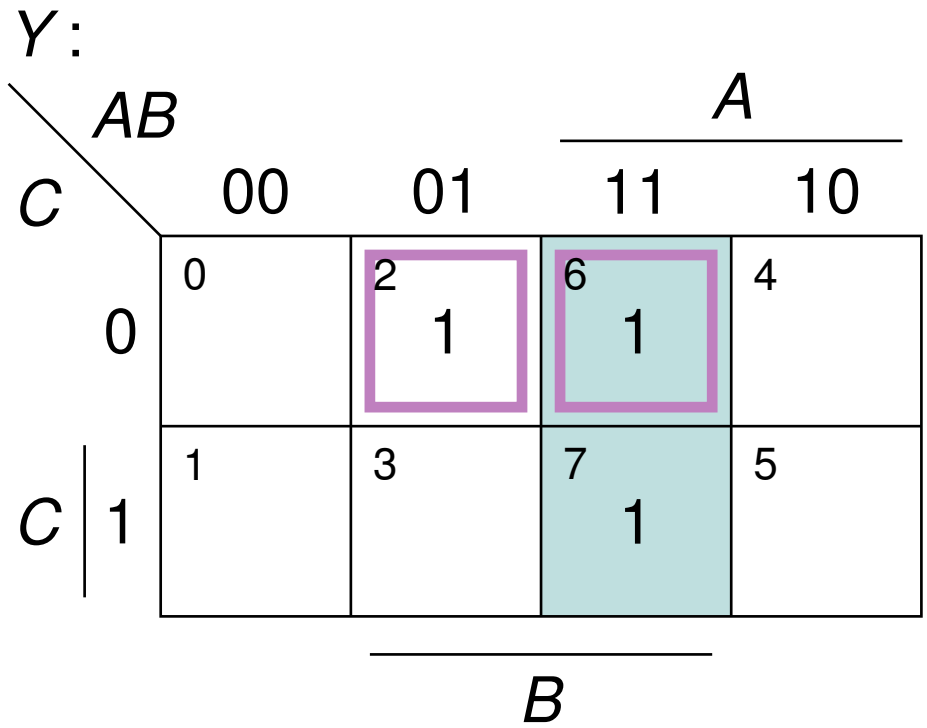
A	B	Y	Minterm
0	0	0	$m_0 = \overline{A}\,\overline{B}$
0	1	0	$m_1 = \overline{A}\,B$
1	0	1	$m_2 = A\,\overline{B}$
1	1	1	$m_3 = A\,B$



Karnaugh Diagramm für drei Eingänge

TECHNISCHE UNIVERSITÄT DARMSTADT

A	B	C	Y	Minterm
0	0	0	0	$m_0 = \overline{A}\,\overline{B}\,\overline{C}$
0	0	1	0	$m_1 = \overline{A}\,\overline{B}\,C$
0	1	0	1	$m_2 = \overline{A}\,B\,\overline{C}$
0	1	1	0	$m_3 = \overline{A}\,B\,C$
1	0	0	0	$m_4 = A\,\overline{B}\,\overline{C}$
1	0	1	0	$m_5 = A\,\overline{B}\,C$
1	1	0	1	$m_6 = A\,B\,\overline{C}$
1	1	1	1	$m_7 = A\,B\,C$

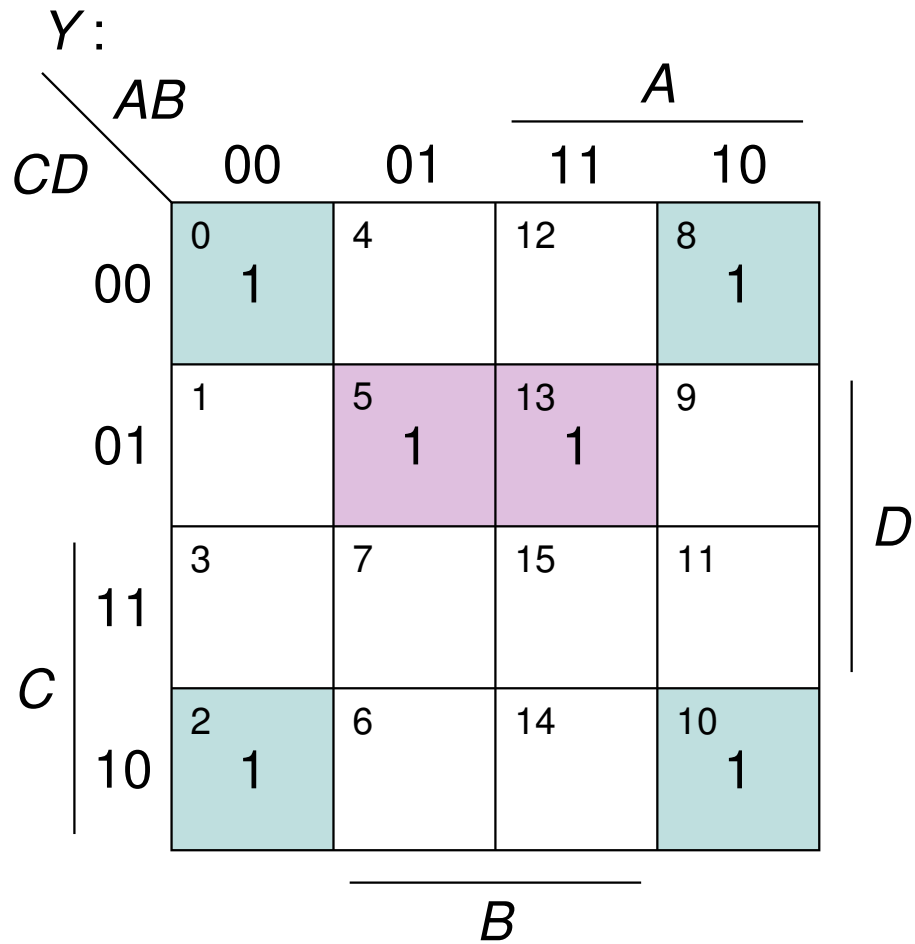


$$Y = A\,B + B\,\overline{C}$$

Karnaugh Diagramm für vier Eingänge

TECHNISCHE UNIVERSITÄT DARMSTADT

A	B	C	D	Y	Minterm
0	0	0	0	1	$m_0 = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$
0	0	0	1	0	$m_1 = \overline{A}\,\overline{B}\,\overline{C}\,D$
0	0	1	0	1	$m_2 = \overline{A}\,\overline{B}\,C\,\overline{D}$
0	0	1	1	0	$m_3 = \overline{A}\,\overline{B}\,C\,D$
0	1	0	0	0	$m_4 = \overline{A}\,B\,\overline{C}\,\overline{D}$
0	1	0	1	1	$m_5 = \overline{A}\,B\,\overline{C}\,D$
0	1	1	0	0	$m_6 = \overline{A}\,B\,C\,\overline{D}$
0	1	1	1	0	$m_7 = \overline{A}\,B\,C\,D$
1	0	0	0	1	$m_8 = A\,\overline{B}\,\overline{C}\,\overline{D}$
1	0	0	1	0	$m_9 = A\,\overline{B}\,\overline{C}\,D$
1	0	1	0	1	$m_{10} = A\,\overline{B}\,C\,\overline{D}$
1	0	1	1	0	$m_{11} = A\,\overline{B}\,C\,D$
1	1	0	0	0	$m_{12} = A\,B\,\overline{C}\,\overline{D}$
1	1	0	1	1	$m_{13} = A\,B\,\overline{C}\,D$
1	1	1	0	0	$m_{14} = A\,B\,C\,\overline{D}$
1	1	1	1	0	$m_{15} = A\,B\,C\,D$



$$Y = \overline{B}\,\overline{D} + B\,\overline{C}\,D$$

Abdeckung von Mintermen durch Implikanten

TECHNISCHE UNIVERSITÄT DARMSTADT

- n Eingangsvariablen
- Implikant aus $k \leq n$ Literalen deckt 2^{n-k} Minterme ab
- Primimplikant
  - nicht vergrößerbare zusammenhängenden viereckigen Fläche im Karnaugh-Diagramm
  - *Achtung:* muss nicht größte Fläche sein

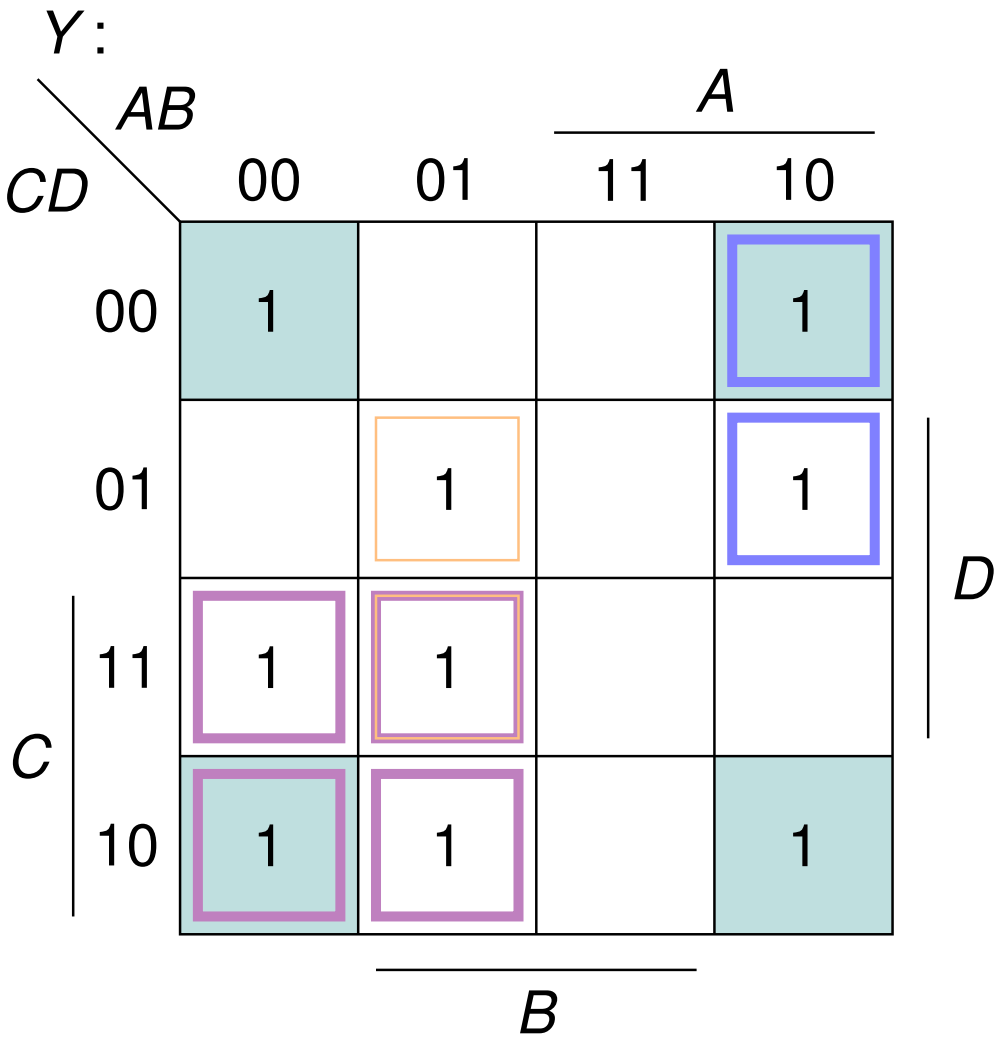
Minimierungsregeln für Karnaugh-Diagramme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Eintragen von Mintermen
  - Einsen aus Wahrheitswertetabelle
  - „Don't Cares“ (*) für ungültige Eingangskombinationen
- Markieren von Implikanten
  - markierte Bereiche dürfen 1 und * enthalten, aber keine 0
  - nur *Rechtecke* mit 2^k Einträgen erlaubt (keine L- oder Z-Formen)
  - Bereiche dürfen sich überschneiden
  - Bereiche dürfen um die Ränder des Diagrammes herum reichen (Torus)
  - Bereiche müssen so groß wie möglich sein (Primimplikanten)
- Ziel: Überdeckung aller Einsen mit möglichst wenigen Primimplikanten

Karnaugh-Diagramm mit vier Eingängen

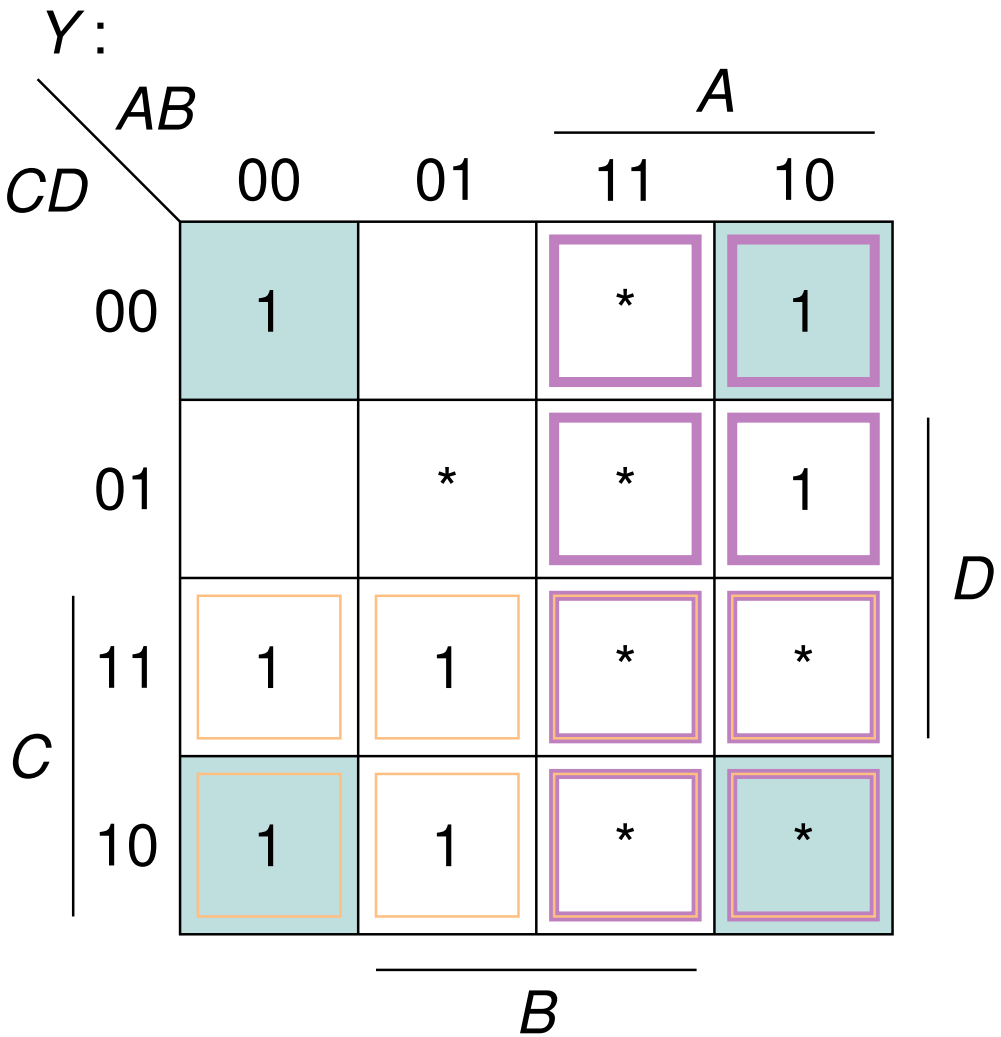
TECHNISCHE UNIVERSITÄT DARMSTADT



$$\overline{B}\,\overline{D} + \overline{A}\,C + \overline{A}\,B\,D + A\,\overline{B}\,\overline{C}$$

Karnaugh-Diagramm mit „Don’t Cares“

TECHNISCHE UNIVERSITÄT DARMSTADT



$\overline{B}\,\overline{D}$
+ A
+ C

Zusammenfassung

TECHNISCHE UNIVERSITÄT DARMSTADT

Zusammenfassung und Ausblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bubble Pushing
- Logik-Realisierung mit Basis-Gattern
- Karnaugh Diagramme

- Nächste Vorlesung behandelt
  - automatisierte Logikminimierung
  - Zeitverhalten von Schaltungen