

Digitaltechnik

Wintersemester 2017/2018

5. Vorlesung



TECHNISCHE
UNIVERSITÄT
DARMSTADT





1. Einleitung
2. Bubble Pushing
3. Logik-Realisierung mit Basis-Gattern
4. Karnaugh Diagramme
5. Zusammenfassung

Einleitung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

11111000000000001111101011100011111011101
01001011110111111101010001110111100000011
0010010111111100010100010100110100110000
1101001011000001001011000100010010101010
1110000100001100001110100110101100101100
1001101101101101100011001000101011001000
1001111101100101100011101110111100111010
1011100001001011110001010100110100010001
1001101101100000111010001100010001000011
1100000111001110101010000101111011100011
0111100000101011011001000100011000011000
1101111100100011011010010100100000101001
0001110001011100001111110011110101011010
0001101100001001011111001111100000101111
1111110110110010001110111111111001111010
0000000011010110010000100000100100000101



- ▶ Anmeldung zu Studienleistung und Fachprüfung offen



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

Rückblick auf die letzten Vorlesungen



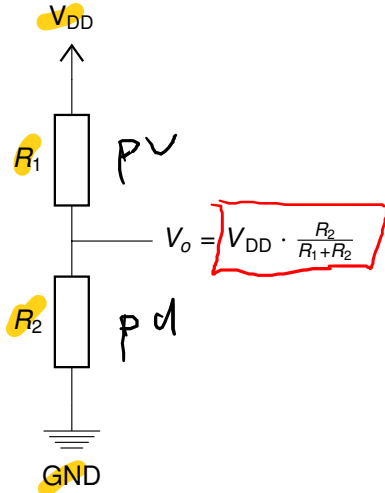
TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ 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



$$R_2 \gg R_1 \Rightarrow V_o \sim V_{DD}$$

$$R_2 \ll R_1 \Rightarrow V_o \sim GND$$

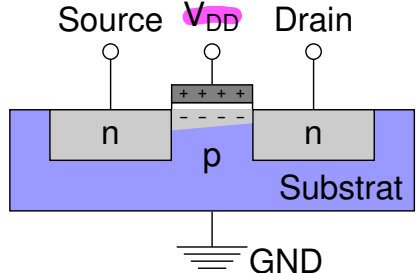
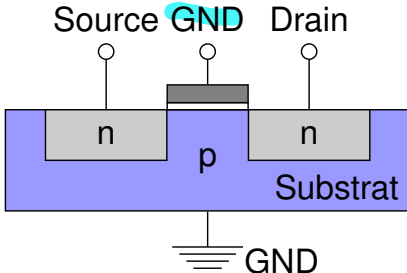
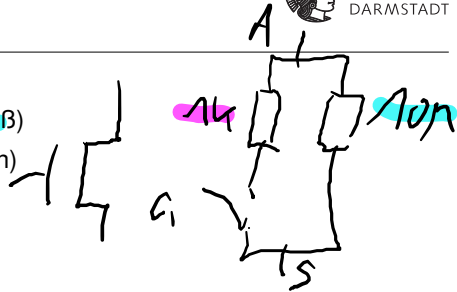
$$R_2 = R_1 \Rightarrow V_o \sim \frac{V_{DD}}{2}$$

Wiederholung: nMOS



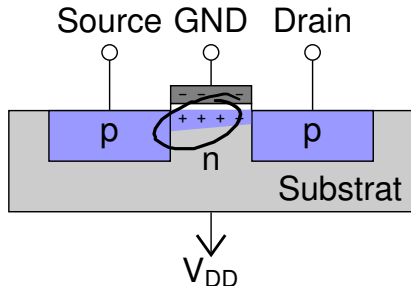
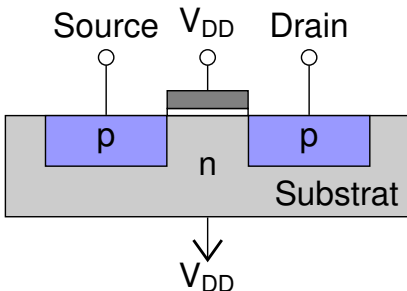
TECHNISCHE
UNIVERSITÄT
DARMSTADT

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



Wiederholung: pMOS

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



Wiederholung: CMOS



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Wiederholung: CMOS

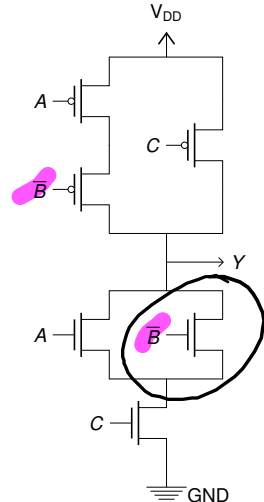
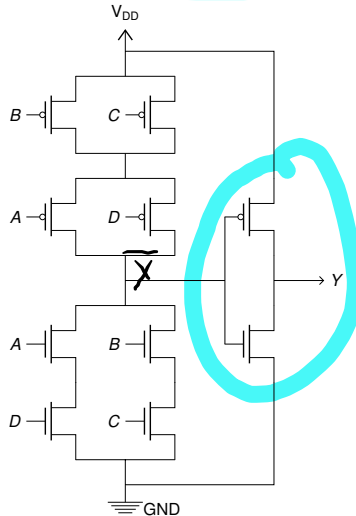
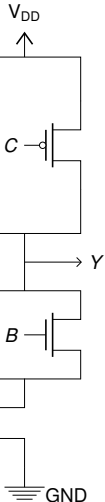


TECHNISCHE
UNIVERSITÄT
DARMSTADT

Wiederholung: CMOS



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Konstruktion von CMOS-Schaltungen



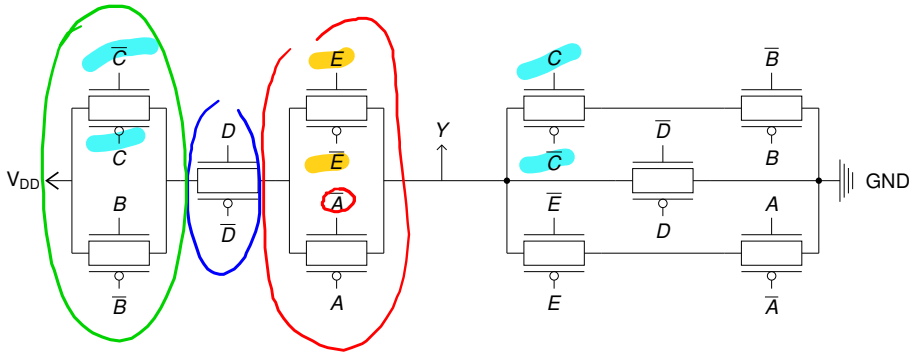
TECHNISCHE
UNIVERSITÄT
DARMSTADT



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

Wiederholung: Transmissionsgatter

$$Y = (E + \bar{A}) \cdot D \cdot (\bar{C} + B)$$



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 $\overline{A \cdot B \cdot C \dots} = \overline{A} + \overline{B} + \overline{C} \dots$	T12' $\overline{A + B + C \dots} = \overline{A} \cdot \overline{B} \cdot \overline{C} \dots$	De Morgan

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



TECHNISCHE
UNIVERSITÄT
DARMSTADT

$$A \cdot B + \bar{A} \cdot C + B \cdot C$$

$$= A \cdot B + \bar{A} \cdot C + 1 \cdot B \cdot C$$

$$= A \cdot B + \bar{A} \cdot C + (A + \bar{A}) \cdot B \cdot C$$

$$= A \cdot B + \bar{A} \cdot C + A \cdot B \cdot C + \bar{A} \cdot B \cdot C$$

$$= A \cdot B + A \cdot B \cdot C + \bar{A} \cdot C + \bar{A} \cdot C \cdot B$$

$$= A \cdot B \cdot 1 + A \cdot B \cdot C + \bar{A} \cdot C \cdot 1 + \bar{A} \cdot C \cdot B$$

$$= A \cdot B \cdot (1 + C) + \bar{A} \cdot C \cdot (1 + B)$$

$$= A \cdot B \cdot 1 + \bar{A} \cdot C \cdot 1$$

$$= A \cdot B + \bar{A} \cdot C$$

Neutralität

Komplement

Distributivität

~~Kommutativität~~

Neutralität

Distributivität

Extremum

Neutralität

q.e.d.

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



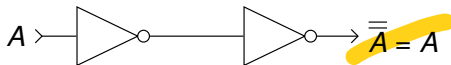
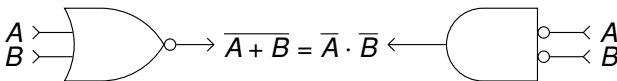
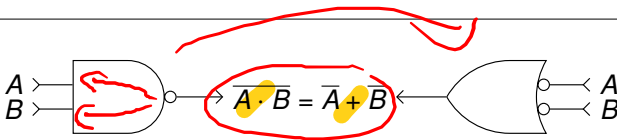
TECHNISCHE
UNIVERSITÄT
DARMSTADT

11111100100110101011010001000010001001100
111010111100011010111111100101010010010001
1011010100000111000111010010101010100011
1100001100010000111000111000000101101000
00111100111100011110001110111110100111100
10000000111100001101111111110000001101101
10000101110100010111111100000011101000110
11011111001000001010010111110101000111
0000100000101000010110001111100101101001
1100011101100010011010000110011101001010
1011111101011000011111010000011100111011
1100010001111010100100011111001101111100
1111010011010011001101111111110010000001
0101011011001111011100110011010100011011
110110000000001111111000010010011111010001
1100100001101010001011101111011011011010

Graphische Umformung von Schaltungen nach De Morgan und Inversion



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Invertierungsblasen verschieben

Bubble Pushing



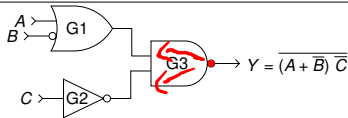
TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ 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



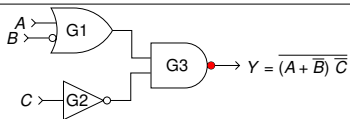
► De Morgan über G3

- Blase am Ausgang → Blase an beiden Eingängen
- AND → OR

Beispiel: Invertierungsblasen rückwärts verschieben

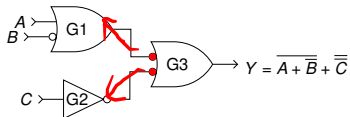


TECHNISCHE
UNIVERSITÄT
DARMSTADT



► De Morgan über G3

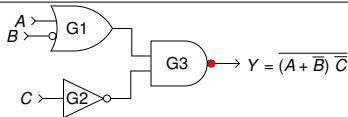
- Blase am Ausgang → Blase an beiden Eingängen
- AND → OR



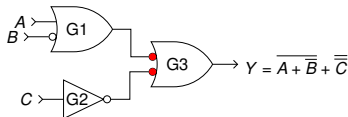
► Blasen entlang Leitungen verschieben

- G3 → G1
- G3 → G2 (Doppelblase aufheben)

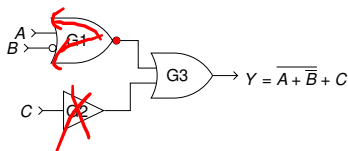
Beispiel: Invertierungsblasen rückwärts verschieben



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

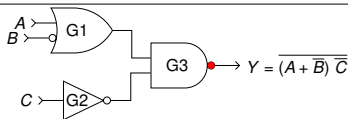


- ▶ Blasen entlang Leitungen verschieben
 - ▶ G3 \rightarrow G1
 - ▶ G3 \rightarrow G2 (Doppelblase aufheben)

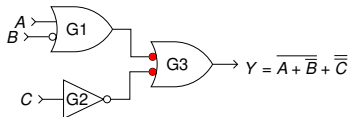


- ▶ De Morgan über G1
 - ▶ Blasen an Ein- und Ausgängen *invertieren*
 - ▶ OR \rightarrow AND
- ▶ Buffer G2 entfernen

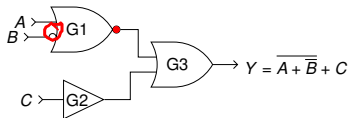
Beispiel: Invertierungsblasen rückwärts verschieben



- ▶ 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



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

Logik-Realisierung mit Basis-Gattern



TECHNISCHE
UNIVERSITÄT
DARMSTADT

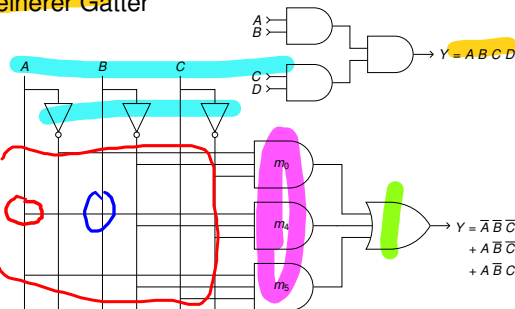
0100001111110011110010001000001111011101
1110010111011110101010001011000100111011
1001000110100101011010111001000101000111
0111000001011111010001100101100011111000
1011011110001111010001010100010110101011
0111110011000111001101001001101001101001
1011001101001110111011011000100111111010
111010110011001001001001110101111101110100
10001100000000010101010011110100010000000
0011101110100001001100110001101101010111
0010100110001001010111101101011010010100
1100110111110010101101100110000110101011
1111000110110010000000101000001110101001
1001001101110101001100111010001100100111
0010010001001100001100100000001011100110
1111111110001010110000010000001000011001

- ▶ 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

⇒ jede boole'sche Funktion realisierbar mit Basisgattern

- ▶ AND2
- ▶ OR2
- ▶ NOT

Handwritten red text: } Minterm

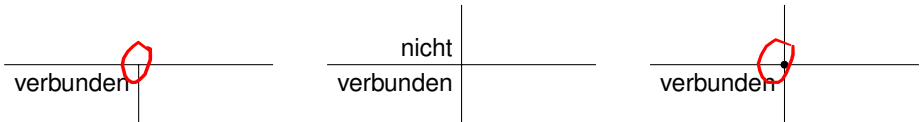


Konventionen für lesbare Schaltpläne



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Eingänge links (oder oberen)
 - ▶ Ausgänge rechts (oder unteren)
 - ▶ Gatter von links nach rechts (oben nach unten) angeordnet
 - ▶ gerade (oder rechtwinklige)
- ⇒ keine Schrägen oder Kurven
- ▶ 3-armige Kreuzungen gelten implizit als verbunden
 - ▶ 4-armige Kreuzungen gelten nur bei Markierung (Punkt) als verbunden





- ▶ 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“)

$$A \oplus B = \overline{A}B + A\overline{B}$$

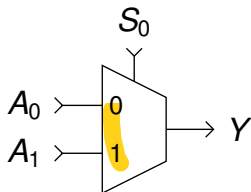
Multiplexer

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



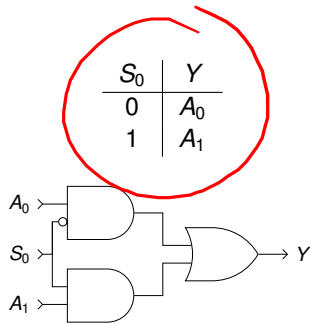
TECHNISCHE
UNIVERSITÄT
DARMSTADT

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



MUX2

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



$$\bar{S}_0 A_0 + S_0 A_1$$

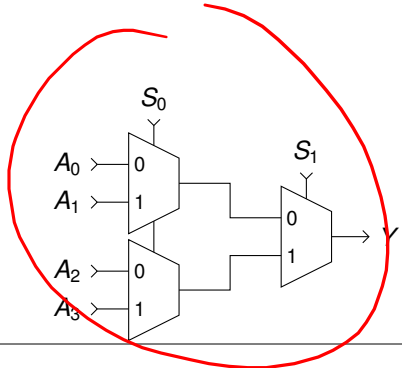
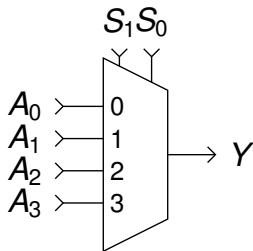
Multiplexer

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



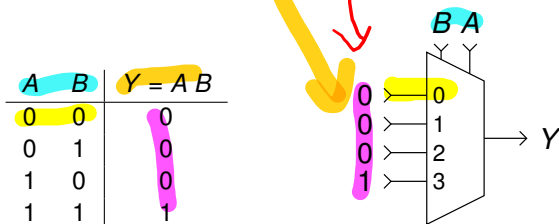
TECHNISCHE
UNIVERSITÄT
DARMSTADT

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



Logikrealisierung mit Multiplexern

- ▶ Variablen als Steuersignale verwenden
- ▶ Wahrheitswertetabelle als Konstanten an Dateneingängen
- ▶ entspricht adressiertem Speicherzugriff
 - ▶ Look-up Tabelle
 - ▶ ROM oder RAM → rekonfigurierbare Logik



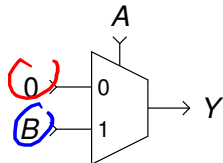
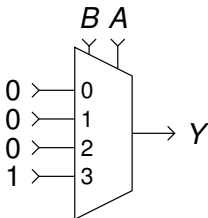
Logikrealisierung mit Multiplexern



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Variablen als Steuersignale verwenden
- ▶ Wahrheitstabelle als Konstanten an Dateneingängen
- ▶ entspricht adressiertem Speicherzugriff
 - ▶ Look-up Tabelle
 - ▶ ROM oder RAM → rekonfigurierbare Logik
- ▶ weitere funktionsspezifische Optimierungen möglich

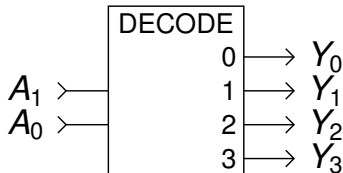
A	B	$Y = A B$
0	0	0
0	1	0
1	0	0
1	1	1





- ▶ n Eingänge A_0, \dots, A_{n-1}
- ▶ 2^n Ausgänge Y_0, \dots, Y_{2^n-1}
- ▶ „One-hot“ Kodierung: $Y_i = u_{2,n}(A_{n-1} \dots A_0) == i ? 1 : 0$

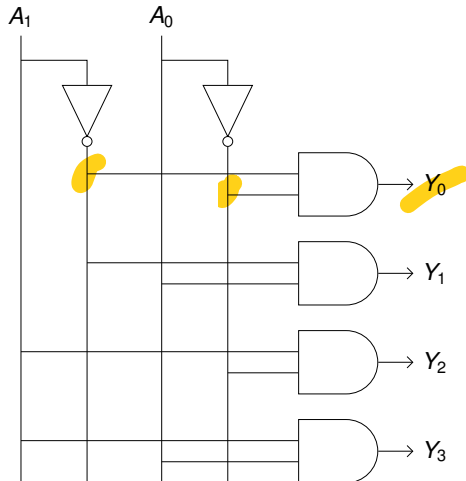
A_1	A_0	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Implementierung von Dekodierern



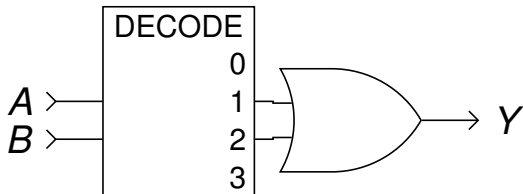
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Logikrealisierung mit Decodern

- ▶ Summe über Minterme, auf denen Zielfunktion wahr ist
- ⇒ 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

0111001100010011101010111000001011101000
0011100000110101101000101101011001100110
1010010001101001101011001100100111010110
1111110100010101000100010010010011011100
1011110010010100111100011101101111000010
1000101100001001010110111100001010110100
0101100010010010001100010011110111100100
00000100000000101001001110101111011000000
10100111111111101101101000011101001001000
01011100111111101110001111011000101100011
1111000000100000010111000001010010000010
1111001011111111100001101010010100010001
1100001101011001101111101010100001000011
111001101001001000000001011100000000101000
0001100110001011111101101110001111001011
1111111001101000000110111010000100110000

Maurice Karnaugh, 1924 -



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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

⇒ Karnaugh(-Veitch) Diagramme



Karnaugh Diagramme

- ▶ Boole'sche Ausdrücke können durch Zusammenfassen von Mintermen minimiert werden

- ▶ $Y = A B + A \bar{B} = A$

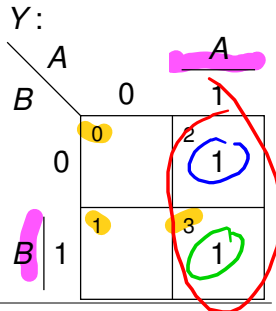
- ▶ Karnaugh-Diagramme stellen Zusammenhänge graphisch dar

- ▶ geschickte Anordnung der Wahrheitswertetabelle
 - ▶ benachbarte Einträge gehören zu gleichem Literal

⇒ Zusammenhängende Minterme besser erkennbar

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

$$Y = A$$



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

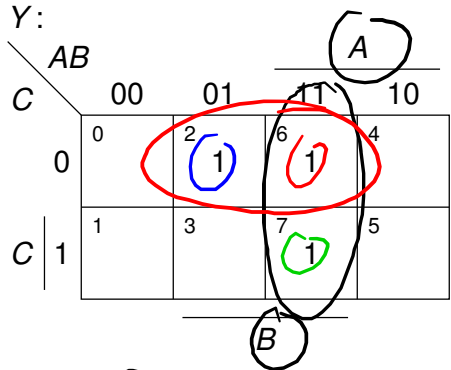
A	B	Y	Minterm
0	0	0	$m_0 = \bar{A} \bar{B}$
0	1	0	$m_1 = \bar{A} B$
1	0	1	$m_2 = A \bar{B}$
1	1	1	$m_3 = A B$

Y:

		A	
		0	1
B	0	0	2 1
	1	1	3 1

Karnaugh Diagramm für drei Eingänge

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



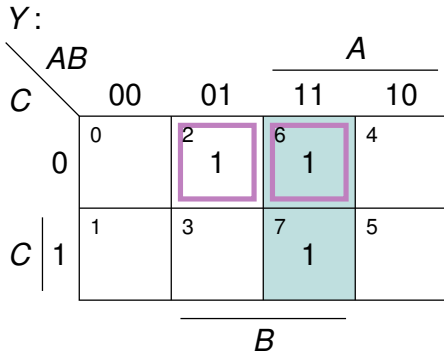
$$Y = AB + B\bar{C}$$

Karnaugh Diagramm für drei Eingänge



TECHNISCHE
UNIVERSITÄT
DARMSTADT

<i>A</i>	<i>B</i>	<i>C</i>	<i>Y</i>	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	$m_7 = A B C$

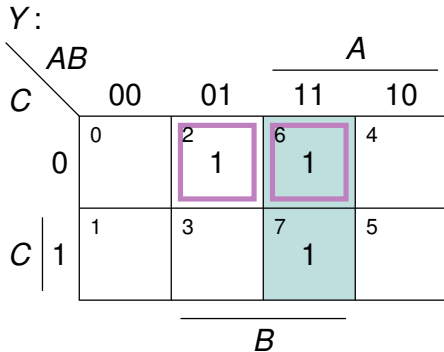


Karnaugh Diagramm für drei Eingänge



TECHNISCHE
UNIVERSITÄT
DARMSTADT

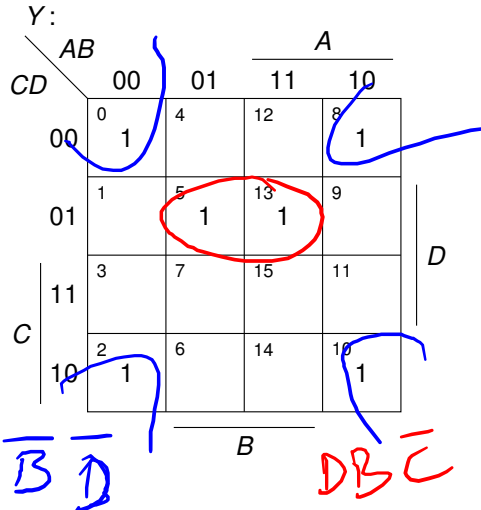
A	B	C	Y	Minterm
0	0	0	0	$m_0 = \overline{A} \overline{B} \overline{C}$
0	0	1	0	$m_1 = \overline{A} \overline{B} C$
0	1	0	1	$m_2 = \overline{A} B \overline{C}$
0	1	1	0	$m_3 = \overline{A} B C$
1	0	0	0	$m_4 = A \overline{B} \overline{C}$
1	0	1	0	$m_5 = A \overline{B} C$
1	1	0	1	$m_6 = A B \overline{C}$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



$$Y = AB + B\overline{C}$$

Karnaugh Diagramm für vier Eingänge

A	B	C	D	Y	Minterm
0	0	0	0	1	$m_0 = \bar{A}\bar{B}\bar{C}\bar{D}$
0	0	0	1	0	$m_1 = \bar{A}\bar{B}\bar{C}D$
0	0	1	0	1	$m_2 = \bar{A}\bar{B}C\bar{D}$
0	0	1	1	0	$m_3 = \bar{A}\bar{B}CD$
0	1	0	0	0	$m_4 = \bar{A}B\bar{C}\bar{D}$
0	1	0	1	1	$m_5 = \bar{A}B\bar{C}D$
0	1	1	0	0	$m_6 = \bar{A}BC\bar{D}$
0	1	1	1	0	$m_7 = \bar{A}BCD$
1	0	0	0	1	$m_8 = A\bar{B}\bar{C}\bar{D}$
1	0	0	1	0	$m_9 = A\bar{B}\bar{C}D$
1	0	1	0	1	$m_{10} = A\bar{B}C\bar{D}$
1	0	1	1	0	$m_{11} = A\bar{B}CD$
1	1	0	0	0	$m_{12} = AB\bar{C}\bar{D}$
1	1	0	1	1	$m_{13} = AB\bar{C}D$
1	1	1	0	0	$m_{14} = ABC\bar{D}$
1	1	1	1	0	$m_{15} = ABCD$



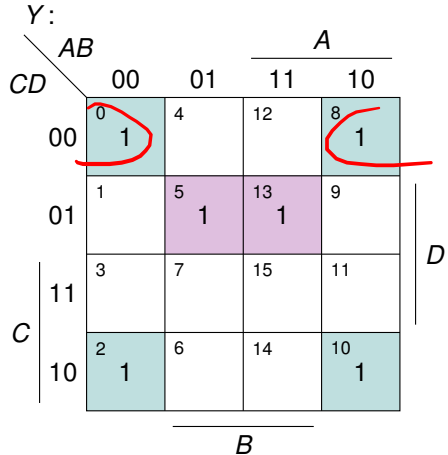
Karnaugh Diagramm für vier Eingänge

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



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	$m_3 = \overline{A}\overline{B}CD$
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	$m_6 = \overline{A}BC\overline{D}$
0	1	1	1	0	$m_7 = \overline{A}BCD$
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	$m_{11} = A\overline{B}CD$
1	1	0	0	0	$m_{12} = AB\overline{C}\overline{D}$
1	1	0	1	1	$m_{13} = AB\overline{C}D$
1	1	1	0	0	$m_{14} = ABC\overline{D}$
1	1	1	1	0	$m_{15} = ABCD$



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



- ▶ 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



- ▶ Eintragen von Mintermen
 - ▶ Einsen aus Wahrheitstabelle
 - ▶ „Don't Cares“ (X) für ungültige Eingangskombinationen
- ▶ Markieren von Implikanten
 - ▶ markierte Bereiche dürfen 1 und X enthalten, aber keine 0
 - ▶ nur Rechtecke mit 2^n 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

Y:

AB		A		
		11	10	
CD	00	01	11	10
	00	1		1
01		1		1
11	1	1		
10	1	1		1

C

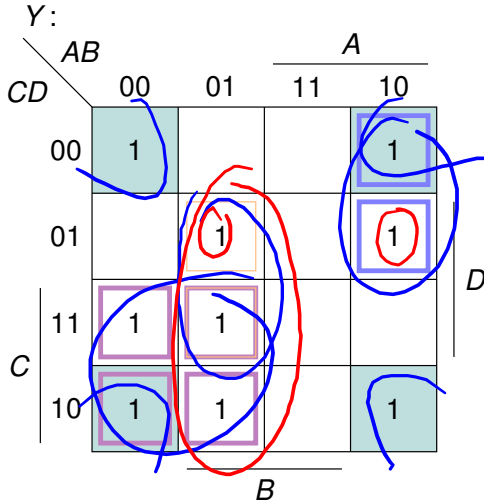
D

B

Karnaugh-Diagramm mit vier Eingängen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

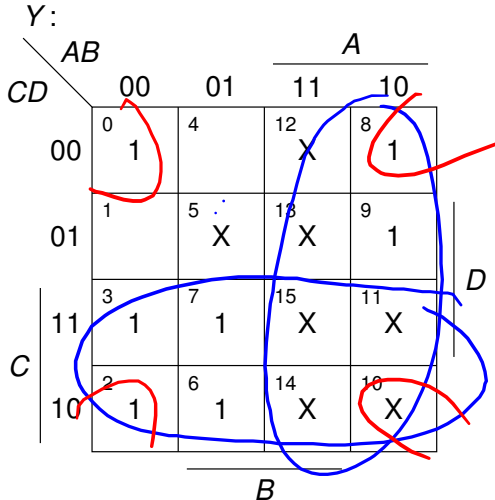


$$\begin{aligned} & \overline{B} \overline{D} \\ + & \overline{A} C \\ + & \overline{A} B D \\ + & A \overline{B} \overline{A} \end{aligned}$$

Karnaugh-Diagramm mit „Don't Cares“



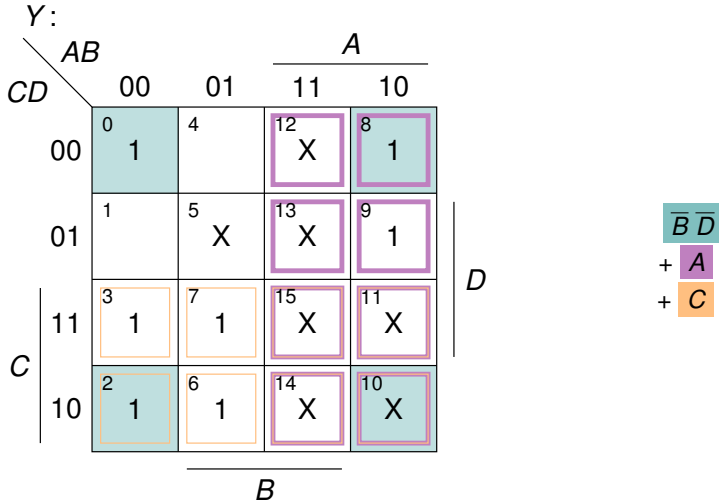
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Karnaugh-Diagramm mit „Don't Cares“



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Zusammenfassung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

0011100011001111001001010001010111011010
1100000010100011101101001100101101111010
0011100110101111010101101000101011001111
1100000000000001000010100110010101111001
0101100000010111100001001111011110100011
0100111001100100000111100101010000101011
1001101010011110000000101111010100010111
0001110010100101000101011011011011101001
11111111000010110001101110011101111001011
01011111101100000000000011011110000101010
00010011110100111100011001111111110101001
1101011001001010001001010100010011010011
01010111001111011110111011010101010110000
10110011100011100000001001000010110111111
1011100011110010001000110110111111111010
110110101010101011101110100011000010100100



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



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

- ▶ Nächste Vorlesung behandelt
 - ▶ automatisierte Logikminimierung
 - ▶ Zeitverhalten von Schaltungen