

Digitaltechnik

Wintersemester 2021/2022

5. Vorlesung



TECHNISCHE
UNIVERSITÄT
DARMSTADT





Umfrage










1. Einleitung

2. Bubble Pushing

3. Logik-Realisierung mit Basis-Gattern

4. Karnaugh Diagramme

5. Zusammenfassung

Anwendungs- software		Programme
Betriebs- systeme		Gerätetreiber
Architektur		Befehle Register
Mikro- architektur		Datenpfade Steuerung
Logik		Addierer Speicher
Digital- schaltungen		UND Gatter Inverter
Analog- schaltungen		Verstärker Filter
Bauteile		Transistoren Dioden
Physik		Elektronen

Agenda



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1. Einleitung

2. Bubble Pushing

3. Logik-Realisierung mit Basis-Gattern

4. Karnaugh Diagramme

5. Zusammenfassung

Anwendungs- software		Programme
Betriebs- systeme		Gerätetreiber
Architektur		Befehle Register
Mikro- architektur		Datenpfade Steuerung
Logik		Addierer Speicher
Digital- schaltungen		UND Gatter Inverter
Analog- schaltungen		Verstärker Filter
Bauteile		Transistoren Dioden
Physik		Elektronen

Überblick der heutigen Vorlesung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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



Harris 2013/2016
Kap. 2.4, 2.5, 2.7, 2.8

Agenda



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1. Einleitung

2. Bubble Pushing

3. Logik-Realisierung mit Basis-Gattern

4. Karnaugh Diagramme

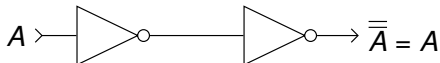
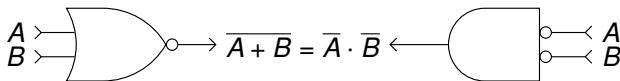
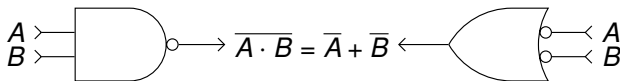
5. Zusammenfassung

Anwendungs- software		Programme
Betriebs- systeme		Gerätetreiber
Architektur		Befehle Register
Mikro- architektur		Datenpfade Steuerung
Logik		Addierer Speicher
Digital- schaltungen		UND Gatter Inverter
Analog- schaltungen		Verstärker Filter
Bauteile		Transistoren Dioden
Physik		Elektronen

Graphische Umformung von Schaltungen nach De Morgan und Involution



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Invertierungsblasen verschieben

Bubble Pushing



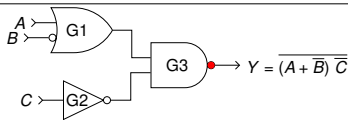
TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ 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 ändern: vorhanden \leftrightarrow nicht vorhanden

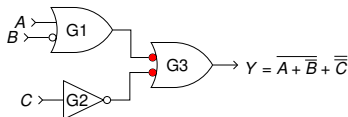
- ▶ zwischen Gattern
 - ▶ vorwärts: Treiber \rightarrow *alle* Empfänger
 - ▶ rückwärts: *alle* Empfänger \rightarrow Treiber
 - ▶ doppelte Blasen heben sich gegenseitig auf (Involution)

- ▶ verbleibende Buffer (vorher Inverter) können entfernt werden

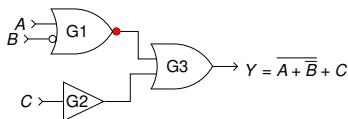
Beispiel: Invertierungsblasen rückwärts verschieben



- ▶ 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 Inverter
 - ▶ weniger Literale (z.B. nur A statt A, \bar{A})
 - ▶ weniger verschiedene Gatter-Arten → einfachere Zellbibliothek (z.B. nur AND, kein OR)
- ▶ Komplementäre Schaltungen für CMOS-Schaltung ableiten
 - ▶ Y für Pull-Up Netzwerk $\leftrightarrow \bar{Y}$ für Pull-Down Netzwerk
 - ▶ $Y = \bar{A}B + C$
 - ▶ $\bar{Y} = \overline{\bar{A}B + C} = \overline{\bar{A}B} \bar{C} = (A + \bar{B})\bar{C}$

Agenda



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1. Einleitung

2. Bubble Pushing

3. Logik-Realisierung mit Basis-Gattern

4. Karnaugh Diagramme

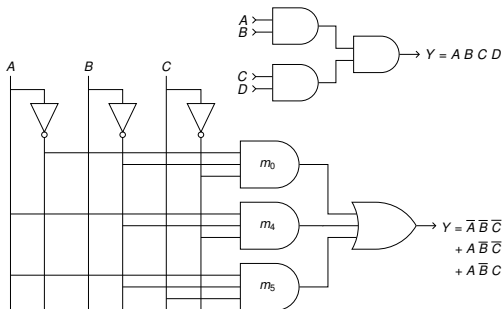
5. Zusammenfassung

Anwendungs- software		Programme
Betriebs- systeme		Gerätetreiber
Architektur		Befehle Register
Mikro- architektur		Datenpfade Steuerung
Logik		Addierer Speicher
Digital- schaltungen		UND Gatter Inverter
Analog- schaltungen		Verstärker Filter
Bauteile		Transistoren Dioden
Physik		Elektronen

- ▶ direkte (konstruktive) Umsetzung der disjunktiven Normalform (DNF)
 - ▶ Eingangsliterale: ein Inverter pro Variable (falls benötigt)
 - ▶ Minterme: je ein “breites” AND Gatter an passende Literale anschließen
 - ▶ Summe: alle Minterme an ein “breites” OR Gatter anschließen
- ▶ Gatter mit vielen Inputs als Bäume kleinerer Gatter

⇒ jede boole'sche Funktion
realisierbar mit Basisgattern

- ▶ AND2
- ▶ OR2
- ▶ NOT

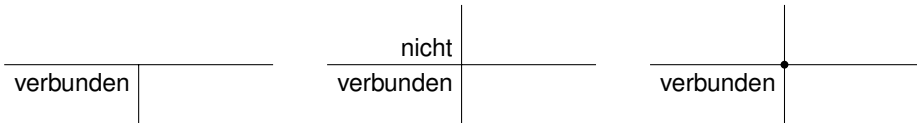


Konventionen für lesbare Schaltpläne



TECHNISCHE
UNIVERSITÄT
DARMSTADT

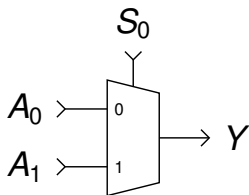
- ▶ Eingänge links (oder oben)
- ▶ Ausgänge rechts (oder unten)
- ▶ Gatter von links nach rechts (oben nach unten) angeordnet
- ▶ gerade (oder rechtwinklige) Verbindungen
- ⇒ keine Schrägen oder Kurven
- ▶ 3-armige Kreuzungen gelten implizit als verbunden
- ▶ 4-armige Kreuzungen gelten nur bei Markierung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



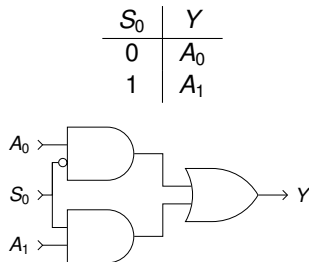


- ▶ zweistufige Logik
 - ▶ sehr mächtig
 - ▶ aufwändige Darstellung und Realisierung
 - ▶ realisiertes Verhalten nicht intuitiv ersichtlich
- ▶ weitere Basisgatter neben AND, OR, NOT:
 - ▶ XOR: Parität
 - ▶ Multiplexer (MUX): n zu 1
 - ▶ Dekodierer (DEC): n zu 2^n

- ▶ Selektiert einen der n Dateneingänge A_0, \dots, A_{n-1} als Ausgang Y
- ▶ $k = \lceil \log_2 n \rceil$ Steuersignale S_0, \dots, S_{k-1}
- ▶ $Y = A_{u_{2,k}(S_{k-1} \dots S_0)}$



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



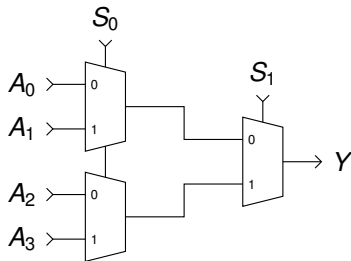
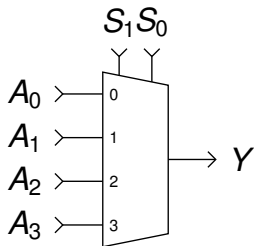
Multiplexer

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

LQ7-4

RQ7-4

MUX1

MUX2

MUX3

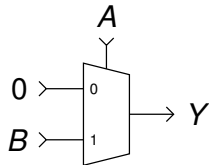
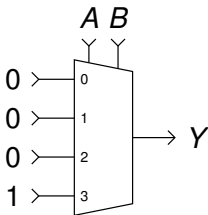
MUX4



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Variablen als Steuersignale verwenden
- ▶ Wahrheitswertetabelle als Konstanten an Dateneingängen
- ▶ entspricht adressiertem Speicherzugriff
 - ▶ Look-up Tabelle
 - ▶ ROM oder RAM → rekonfigurierbare Logik
- ▶ Beliebige Funktion mit N Variablen kann sogar via $\text{MUX}2^{N-1}$ realisiert werden (s. Harris, Fig. 2.60)

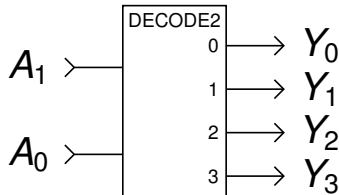
A	B	$Y = A B$
0	0	0
0	1	0
1	0	0
1	1	1





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

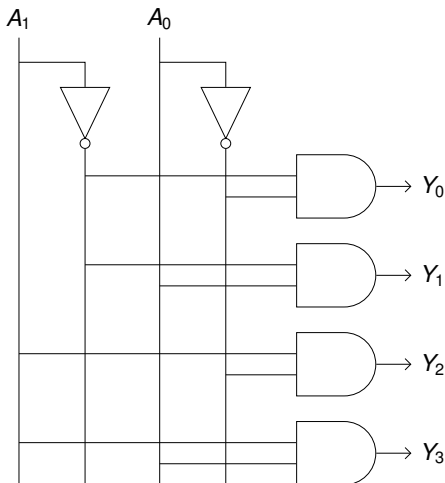
A_1	A_0	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Implementierung von Dekodierern



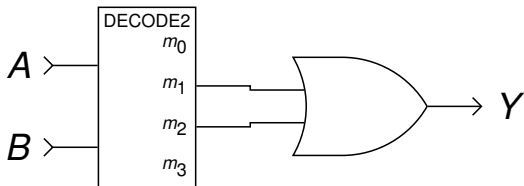
TECHNISCHE
UNIVERSITÄT
DARMSTADT





- Summe über Minterme, auf denen Zielfunktion wahr ist
- ⇒ Decoder ersetzt erste Stufe der zweistufigen Logikrealisierung

<i>A</i>	<i>B</i>	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0





Umfrage

Agenda



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1. Einleitung

2. Bubble Pushing

3. Logik-Realisierung mit Basis-Gattern

4. Karnaugh Diagramme

5. Zusammenfassung

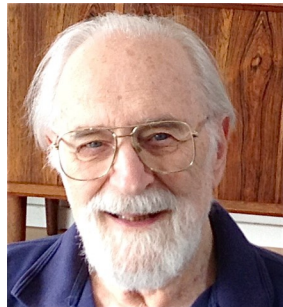
Anwendungs- software		Programme
Betriebs- systeme		Gerätetreiber
Architektur		Befehle Register
Mikro- architektur		Datenpfade Steuerung
Logik		Addierer Speicher
Digital- schaltungen		UND Gatter Inverter
Analog- schaltungen		Verstärker Filter
Bauteile		Transistoren Dioden
Physik		Elektronen

Maurice Karnaugh, 1924 -



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Bell Labs
 - ▶ IBM Research
 - ▶ Techniken und Methoden für den schnellen Entwurf informationstechnischer Systeme
- ⇒ Karnaugh(-Veitch) Diagramme



Graycode



TECHNISCHE
UNIVERSITÄT
DARMSTADT



- ▶ boole'sche Ausdrücke können durch Zusammenfassen von Mintermen minimiert werden
 - ▶ $Y = A B + A \bar{B} = A$
- ▶ Karnaugh Diagramme stellen Zusammenhänge graphisch dar
 - ▶ Anordnung der Wahrheitwertetabelle via Gray code
 - ⇒ 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:

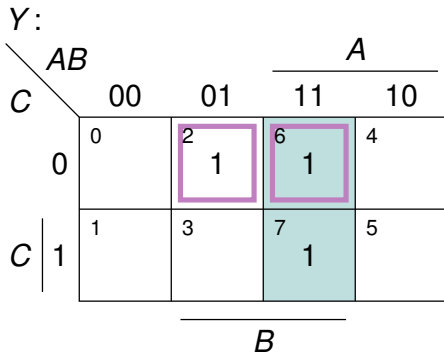
		A	
		0	1
B	0	0	2 1
	1	1	3 1

Karnaugh Diagramm für drei Eingänge **LQ4-2**



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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	$m_7 = A B C$



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



- ▶ n Eingangsvariablen
- ▶ Implikant aus $k \leq n$ Literalen deckt 2^{n-k} Minterme ab
- ▶ Primimplikant
 - ▶ nicht vergrößerbare zusammenhängende viereckige Fläche im Karnaugh Diagramm
 - ▶ *Achtung:* Umbruch an Rändern beachten

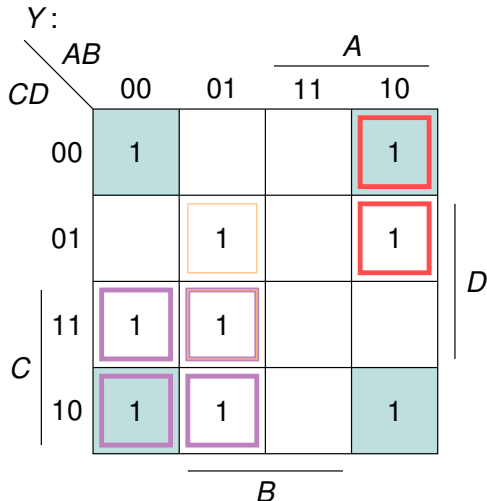


- ▶ Eintragen von Mintermen
 - ▶ Einsen aus Wahrheitstabelle
 - ▶ “Don’t Cares” (*) für ungültige Eingangskombinationen
- ▶ Markieren von Implikanten
 - ▶ markierte Bereiche dürfen 1 und * enthalten, aber keine 0
 - ▶ nur *Rechtecke* mit 2^k Einträgen erlaubt (keine L- oder Z-Formen)
 - ▶ Bereiche dürfen 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



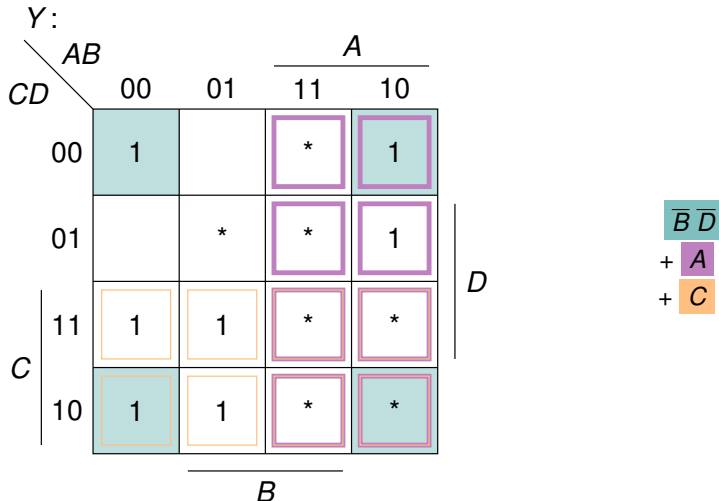
$$\begin{aligned} & \bar{B} \bar{D} \\ + & \bar{A} C \\ + & \bar{A} B D \\ + & A \bar{B} \bar{C} \end{aligned}$$

Karnaugh Diagramm mit "Don't Cares" LQ4-4

RQ4-4



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Agenda



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1. Einleitung

2. Bubble Pushing

3. Logik-Realisierung mit Basis-Gattern

4. Karnaugh Diagramme

5. Zusammenfassung

Anwendungs- software		Programme
Betriebs- systeme		Gerätetreiber
Architektur		Befehle Register
Mikro- architektur		Datenpfade Steuerung
Logik		Addierer Speicher
Digital- schaltungen		UND Gatter Inverter
Analog- schaltungen		Verstärker Filter
Bauteile		Transistoren Dioden
Physik		Elektronen



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

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

DT Präsenzbearbeitung

Fr, 11:40-13:20 in S103/226 (105 Plätze bei 50% Belegung, **derzeit 3G Nachweis**)

	Anmeldungen (max. 100)	Anwesend
Fr, 22.10.	100	ca. 70
Fr, 29.10.	92	ca. 80
Fr, 05.11.	73	ca. 55
Fr, 12.11.	45	ca. 40

DT Hybrid-Vorlesung?

Mi, 9:50-11:30; Raum mit 242 Plätzen bei 50% Belegung ist geblockt

Umfrage: Wer von Ihnen hätte Stand JETZT prinzipiell Interesse an einer DT Hybrid-**Vorlesung** und würde in den Hörsaal kommen wollen?



- ▶ **Termin: 08.03.2022, 11:00 - 13:00 Uhr**
 - ▶ Bearbeitungszeit: 90 Minuten
 - ▶ **Derzeit in Präsenz geplant.** Raumzuteilung wird rechtzeitig bekannt gegeben
- ▶ **Anmeldung in TUCaN bis 31.01.2022 notwendig**
 - ▶ Für **Fachprüfung** (ab 01.11.2021) und **ggf. Studienleistung** (ab 01.09.2021)
 - ▶ **Am besten direkt anmelden!**
- ▶ Kein Notenbonus nach §25(2) APB
- ▶ Keine Hilfsmittel (Taschenrechner, etc.) zugelassen
 - ▶ Ausnahme: Hilfsblatt
- ▶ Klausuraufbau: je Themenblock 1/3 der Punkte für
 - ▶ Wissens- und Verständnisfragen (s. Vorlesungen)
 - ▶ Übungsaufgaben (s. Übungsblätter)
 - ▶ Transferaufgaben
- ▶ Wiederholungsklausur im Herbst 2022
- ▶ Wir stellen keine alten Klausuren zur Verfügung