

# Digitaltechnik

## Wintersemester 2021/2022

### 6. Vorlesung



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT





# Umfrage zur letzten Woche



- ▶ **Termin: Dienstag, 08.03.2022, 11:00 - 13:00 Uhr**
  - ▶ Bearbeitungszeit: 90 Minuten
  - ▶ **Derzeit in Präsenz geplant.** Raumzuteilung wird rechtzeitig bekannt gegeben
- ▶ Wiederholungsklausur im Herbst 2022
- ▶ Wir stellen keine alten Klausuren zur Verfügung

DT Präsenzbearbeitung		Interesse Präsenzteilnahme bei DT Hybrid-Vorlesung			
Datum	Anmeldungen / Anwesende	Datum	100%	50%	0%
Fr, 22.10.	70 / 100				
Fr, 29.10.	80 / 92	Mi, 03.11.	101	98	10
Fr, 05.11.	55 / 73	Mi, 10.11.	69	87	89
Fr, 12.11.	40 / 45	Mi, 17.11.	50	60	84
Fr, 19.11.	25 / 26				

Aufgrund der aktuellen Pandemielage und Vorgaben (3G im Hörsaal) möchten wir die DT Präsenzbearbeitung derzeit beibehalten und die Planung zur zusätzlichen DT Hybrid-Vorlesung vorerst stoppen.

**Bitte passen Sie gut auf sich selbst und auf Andere auf!**










## 1. Einleitung

## 2. Algorithmische Logikminimierung

## 3. Mehrwertige Logik

## 4. Zeitverhalten

## 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. Algorithmische Logikminimierung

## 3. Mehrwertige Logik

## 4. Zeitverhalten

## 5. Zusammenfassung

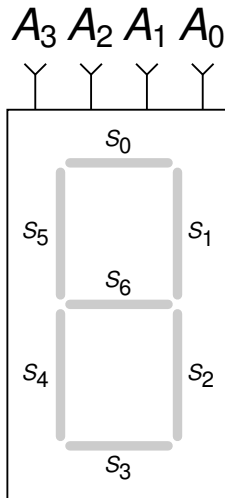
Anwendungs- software		Programme
Betriebs- systeme		Gerätetreiber
Architektur		Befehle Register
Mikro- architektur		Datenpfade Steuerung
Logik		Addierer Speicher
Digital- schaltungen		UND Gatter Inverter
Analog- schaltungen		Verstärker Filter
Bauteile		Transistoren Dioden
Physik		Elektronen

## 7-Segment Anzeige: $\mathbb{B}^4 \rightarrow \mathbb{B}^7$



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

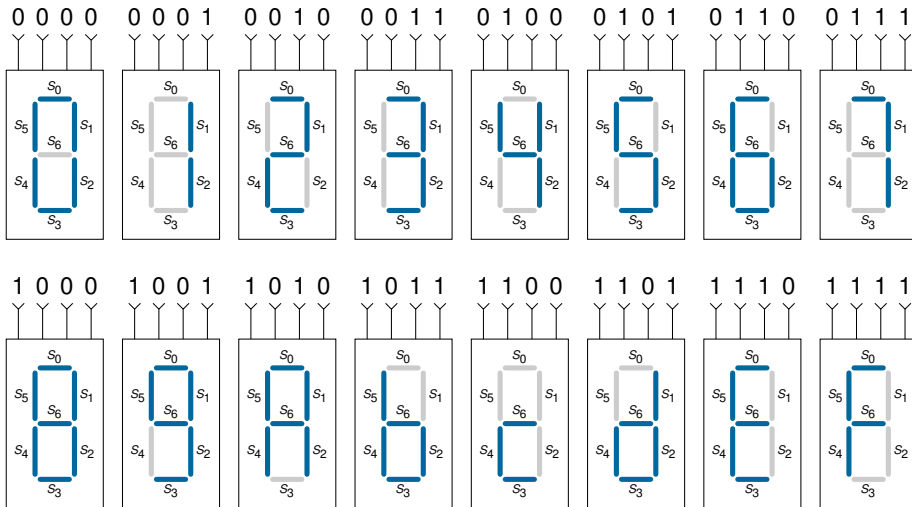
- ▶ (Typ.) vier Eingänge für dargestellte Ziffer
- ▶ Sieben *unabhängig* schaltbare Segmente  $S_0, \dots, S_6$
- ⇒ jedes Segment nur für bestimmte Zeichen aktiv



# Hexadezimale 7-Segment Anzeige (0, ... , F)



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT





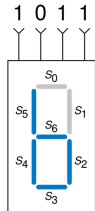
# Hexadezimale 7-Segment Anzeige

## Wahrheitswertetabelle



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	1	1	1	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1
1	1	0	0	0	0	0	1	1	0	1
1	1	0	1	0	1	1	1	1	0	1
1	1	1	0	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	1	1	1



# Hexadezimale 7-Segment Anzeige

## Normalformen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

$$\begin{aligned} \text{DNF: } S_0 &= \overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0} & [m_0] \\ &+ \overline{A_3} \overline{A_2} A_1 \overline{A_0} & [m_2] \\ &+ \overline{A_3} \overline{A_2} A_1 A_0 & [m_3] \\ &+ \overline{A_3} A_2 \overline{A_1} A_0 & [m_5] \\ &+ \overline{A_3} A_2 A_1 \overline{A_0} & [m_6] \\ &+ \overline{A_3} A_2 A_1 A_0 & [m_7] \\ &+ A_3 \overline{A_2} \overline{A_1} \overline{A_0} & [m_8] \\ &+ A_3 \overline{A_2} \overline{A_1} A_0 & [m_9] \\ &+ A_3 \overline{A_2} A_1 \overline{A_0} & [m_{10}] \\ &+ A_3 A_2 A_1 \overline{A_0} & [m_{14}] \\ &+ A_3 A_2 A_1 A_0 & [m_{15}] \end{aligned}$$

$$\begin{aligned} \text{KNF: } S_0 &= (A_3 + A_2 + A_1 + \overline{A_0}) & [M_1] \\ &\cdot (A_3 + \overline{A_2} + A_1 + A_0) & [M_4] \\ &\cdot (\overline{A_3} + A_2 + \overline{A_1} + \overline{A_0}) & [M_{11}] \\ &\cdot (\overline{A_3} + \overline{A_2} + A_1 + A_0) & [M_{12}] \\ &\cdot (\overline{A_3} + \overline{A_2} + A_1 + \overline{A_0}) & [M_{13}] \end{aligned}$$

# Hexadezimale 7-Segment Anzeige

## Verkürzte Minterm/Maxterm-Schreibweise



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

► Boole'sche Funktion eindeutig spezifiziert durch

- Indizes der 1-Minterme für DNF bzw.
- Indizes der 0-Maxterme für KNF

⇒ Erlaubt kompaktere Schreibweise

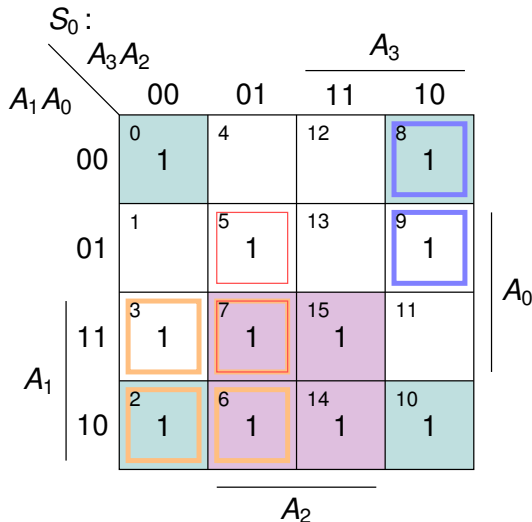
$$\begin{aligned} S_0 &= m_0 + m_2 + m_3 + m_5 + m_6 + m_7 + m_8 + m_9 + m_{10} + m_{14} + m_{15} \\ &= \sum m(0, 2, 3, 5, 6, 7, 8, 9, 10, 14, 15) \\ &= M_1 M_4 M_{11} M_{12} M_{13} \\ &= \prod M(1, 4, 11, 12, 13) \end{aligned}$$

# Hexadezimale 7-Segment Anzeige

## Karnaugh Diagramm



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



$$\begin{aligned}
 S_0 &= \overline{A_0} \overline{A_2} \\
 &+ A_1 A_2 \\
 &+ A_1 \overline{A_3} \\
 &+ A_0 A_2 \overline{A_3} \\
 &+ \overline{A_1} \overline{A_2} A_3
 \end{aligned}$$

# Dezimale 7-Segment Anzeige (0, ... , 9)

## Wahrheitstabelle mit Don't Cares



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

$A_3$	$A_2$	$A_1$	$A_0$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	*	*	*	*	*	*	*
1	0	1	1	*	*	*	*	*	*	*
1	1	0	0	*	*	*	*	*	*	*
1	1	0	1	*	*	*	*	*	*	*
1	1	1	0	*	*	*	*	*	*	*
1	1	1	1	*	*	*	*	*	*	*

# Dezimale 7-Segment Anzeige

## Minterm/Maxterm-Schreibweise mit Don't Cares



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Don't Cares können als 0 oder 1 realisiert werden
- ⇒ in DNF und KNF gleichermaßen enthalten

$$\begin{aligned} S_0 &= m_0 + m_2 + m_3 + m_5 + m_6 + m_7 + m_8 + m_9 + d_{10} + d_{11} + d_{12} + d_{13} + d_{14} + d_{15} \\ &= \sum m(0, 2, 3, 5, 6, 7, 8, 9) + \sum d(10, 11, 12, 13, 14, 15) \\ &= M_1 M_4 D_{10} D_{11} D_{12} D_{13} D_{14} D_{15} \\ &= \prod M(1, 4) \prod D(10, 11, 12, 13, 14, 15) \end{aligned}$$

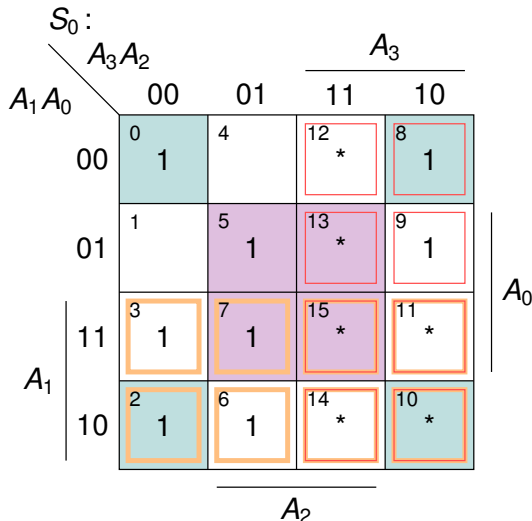
- ▶ *Achtung:* Don't cares nur für verkürzte Schreibweise in einem Ausdruck möglich (kein Teil von DNF oder KNF)

# Dezimale 7-Segment Anzeige

## Karnaugh Diagramm mit Don't Cares



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



$$\begin{aligned}
 S_0 &= \overline{A_0} \overline{A_2} \\
 &+ A_0 A_2 \\
 &+ A_1 \\
 &+ A_3
 \end{aligned}$$

# Dezimale 7-Segment Anzeige

## Karnaugh Diagramm mit Maxtermen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

$S_0$ :

$A_3 A_2$		$A_3$			
		00	01	11	10
$A_1 A_0$	00	0	4 0	12 *	8
	01	1 0	5	13 *	9
	11	3	7	15 *	11 *
	10	2	6	14 *	10 *
		$A_2$			

$A_0$

$$\begin{aligned} \overline{S_0} &= \overline{A_0} \overline{A_1} A_2 \\ &+ A_0 \overline{A_1} \overline{A_2} \overline{A_3} \end{aligned}$$

$$\begin{aligned} S_0 &= (A_0 + A_1 + \overline{A_2}) \\ &\cdot (\overline{A_0} + A_1 + A_2 + A_3) \end{aligned}$$



# Überblick der heutigen Vorlesung



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Algorithmische Logikminimierung
- ▶ Mehrwertige Logik
- ▶ Zeitverhalten



Harris 2013/2016  
Kap. 2.6, 2.9

Katz 2005  
Kap. 3.2

# Agenda



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## 1. Einleitung

## 2. Algorithmische Logikminimierung

## 3. Mehrwertige Logik

## 4. Zeitverhalten

## 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



- ▶ Algebraisch:
  - ▶ Umformen nach Axiomen/Theoremen
- ▶ Grafisch:
  - ▶ Karnaugh Diagramme
- ▶ Algorithmisch
  - ▶ exakt: Quine-McCluskey
  - ▶ heuristisch: Espresso

⇒ Minimiere Anzahl der zur Darstellung einer Funktion notwendigen Implikanten



- ▶ Grafische Verfahren:
  - ▶ für viele ( $> 6$ ) Eingänge nicht mehr praktikabel
  - ▶ keine Optimierung zwischen Ausdrücken für mehrere Ausgänge
- ▶ Quine-McCluskey-Methode
  - ▶ berechnet zunächst *alle* möglichen Implikanten
  - ▶ ermittelt *danach* minimale Teilmenge für vollständige Überdeckung
  - ⇒ Rechenzeit steigt exponentiell in der Anzahl der Eingänge
- ⇒ für wirklich große Probleme ( $> 50$  Variablen) nur Heuristiken sinnvoll
  - ▶ geringere Laufzeitkomplexität
  - ▶ geringere Lösungsqualität

- ▶ in 1980er Jahren von IBM und UC Berkeley entwickelt
- ▶ unterstützt auch mehrere (zusammen optimierte) Ausgänge
- ▶ Details des Algorithmus hier nicht relevant (Buch v. Katz 2005, sowie Rudell 1986 "Multiple-Valued Logic Minimization for PLA Synthesis")
- ▶ hier nur Anwendung einer konkreten Implementierung
  - ▶ <https://embedded.eecs.berkeley.edu/pubs/downloads/espresso>
  - ▶ Anleitung / Quellen auch in Moodle verfügbar
  - ▶ spezielles Dateiformat für boole'sche Funktionen
  - ▶ erlaubt auch exakte Minimierung (als Referenz für Heuristik):  
`espresso -D exact input.esp > output.esp`  
`espresso -D ESPRESSO input.esp > output.esp`

xor.esp

```
1  # Y = A xor B
2  .i    2    # Anzahl Eingänge
3  .o    1    # Anzahl Ausgänge
4  .ilb A B   # Namen der Eingänge (optional)
5  .ob  Y     # Namen der Ausgänge (optional)
6  .p    4    # Anzahl der Tabellenzeilen (optional)
7  00 0      # Ausgang 0 optional
8  01 1
9  10 1
10 11 0      # Ausgang 0 optional
11 .e        # Ende (optional)
```

- ▶ relevante Informationen zeilenweise nach Keywords
  - .i Anzahl  $n_i$  der Eingänge (erforderlich)
  - .o Anzahl  $n_o$  der Ausgänge (erforderlich)
  - .ilb Name(n) der Eingänge
  - .ob Name(n) der Ausgänge
  - .p Anzahl der Tabellenzeilen
  - .e Dateieneinde
- ▶ Wahrheitswertetabelle
  - ▶ jede Zeile beschreibt einen Implikanten mit  $n_i$  Zeichen ...
    - 0 Eingang negiert im Implikanten
    - 1 Eingang nicht-negiert im Implikanten
    - Eingang nicht im Implikanten (kein Minterm)
  - ▶ ... und  $n_o$  Ausgangsfunktionen mit je einem Zeichen
    - 0 Implikant im off set des Ausgangs (optional)
    - 1 Implikant im on set des Ausgangs
    - Implikant im on set oder off set des Ausgangs (Don't Care)
- ▶ “#” leitet Kommentar ein

# Espresso Dezimale 7-Segment Anzeige

## Eingabedateien



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

sevenseg/s0.esp

```
1 # S0 of 7-segment display
2 .i 4
3 .o 1
4 0000 1
5 0010 1
6 0011 1
7 0101 1
8 0110 1
9 0111 1
10 1000 1
11 1001 1
12 1010 -
13 1011 -
14 1100 -
15 1101 -
16 1110 -
17 1111 -
```

sevenseg/all.esp

```
1 # 7-segment display
2 .i 4
3 .o 7
4 0000 1111110
5 0001 0110000
6 0010 1101101
7 0011 1111001
8 0100 0110011
9 0101 1011011
10 0110 1011111
11 0111 1110000
12 1000 1111111
13 1001 1111011
14 1010 -----
15 1011 -----
16 1100 -----
17 1101 -----
18 1110 -----
19 1111 -----
```



# Espresso Dezimale 7-Segment Anzeige Ausgabedateien



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

```
espresso -D ESPRESSO sevenseg/s0.esp
```

```
1  # S0 of 7-segment display
2  .i 4
3  .o 1
4  .p 4
5  -0-0 1
6  1--- 1
7  --1- 1
8  -1-1 1
9  .e
```

$$\overline{A_2} \overline{A_0} + A_3 + A_1 + A_2 A_0$$

```
espresso -D ESPRESSO sevenseg/all.esp
```

```
1  # 7-segment display
2  .i 4
3  .o 7
4  .p 9
5  -0-0 1001100
6  -0-1 0110000
7  --10 1001100
8  -01- 0101001
9  -1-0 0010011
10 --11 1110000
11 --00 0110010
12 -101 1011011
13 1--- 1001011
14 .e
```



- ▶ Mehrwertige Logik
- ▶ Optimierung von Zustandsautomaten
  - ▶ Reduktion der Anzahl der Zustände
  - ▶ Erkennung von äquivalenten Zuständen
  - ▶ Optimierungen der Zustandskodierung



# Pause & Umfrage bis hier

# Agenda



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT










1. Einleitung

2. Algorithmische Logikminimierung

3. Mehrwertige Logik

4. Zeitverhalten

5. Zusammenfassung

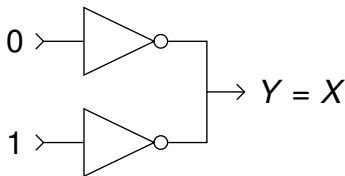
Anwendungs- software		Programme
Betriebs- systeme		Gerätetreiber
Architektur		Befehle Register
Mikro- architektur		Datenpfade Steuerung
Logik		Addierer Speicher
Digital- schaltungen		UND Gatter Inverter
Analog- schaltungen		Verstärker Filter
Bauteile		Transistoren Dioden
Physik		Elektronen

- ▶ bisher galt:
  - ▶ jeder Schaltungsknoten (außer Eingänge) wird von *genau einem* Schaltungselement auf 0 oder 1 getrieben
  - ▶ Axiome der boole'schen Algebra basieren auf  $\mathbb{B} = \{0, 1\}$
- ⇒ ignoriert wichtige Teile der Realität
  - ▶ Wie breiten sich ungültige Spannungen in Schaltung aus?
- ⇒ Unterscheidung von zwei weiteren Logikwerten neben 0 und 1
  - X mehrfach getrieben: fehlerhaft
  - Z ungetrieben/hochohmig (high impedance): gezielt
- ▶ *Achtung:*
  - ▶ nicht mit "Don't Care" (\*) verwechseln
  - ▶ tatsächliche Spannung *kann* auch im 0- oder 1-Bereich liegen, das Schaltungsdesign stellt dies aber nicht sicher

# X (mehrfach getrieben) bei konkurrierenden Ausgängen



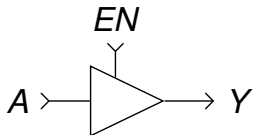
- ▶ mehrere (unabhängige) Treiber für den selben Schaltungsknoten
  - ▶ Konflikt, sobald Treiber in entgegengesetzte Richtung ziehen
    - ▶ instabil: abhängig von Betriebsspannung, Temperatur, etc.
    - ▶ destruktiv: Kurzschluss verursacht hohen Energieverbrauch
  - ▶ fast immer ein Entwurfsfehler
    - ▶ z.B. doppelte Zuweisung in Hardwarebeschreibung
- ⇒ Konflikt-Quelle muss in Simulation leicht nachvollziehbar sein



# Z (ungetrieben/hochohmig) bei Tristate-Buffer



- ▶ zusätzliches Enable-Signal EN an Buffer
  - ▶ EN=1: Funktion wie normaler Buffer
  - ▶ EN=0: Ausgang hochohmig (offen, ungetrieben, floating, high-impedance) Z
- ▶ *Achtung:*  $Z \neq 0$



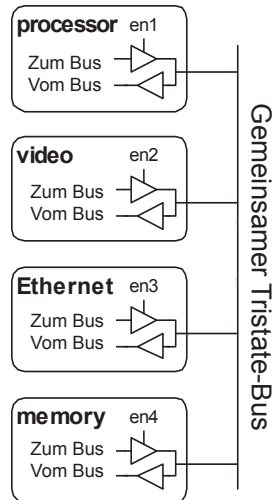
EN	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

# Bus mit Tristate-Buffern



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ mehrere Treiber an gemeinsamer Leitung
- ▶ zu jedem Zeitpunkt *genau ein* aktiver Treiber
- ▶ erlaubt Wechsel der Kommunikationsrichtung

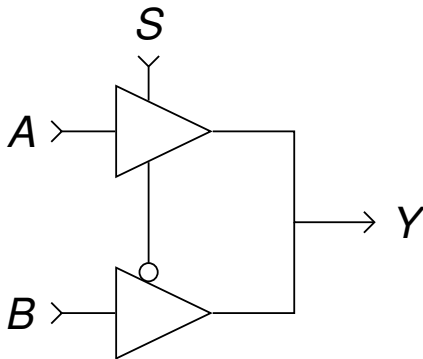




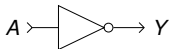
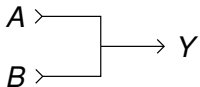
# Tristate-Buffer für Multiplexer



<i>S</i>	<i>A</i>	<i>B</i>	<i>Y</i>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



- ▶ Resolutionstabellen definieren Ausbreitung von X (mehrfach getrieben) und Z (hochohmig)
- ▶ mehr Konvention (für Simulator) als physikalische Realität
- ▶ z.B. IEEE 1164:



A/B	X	0	1	Z
X	X	X	X	X
0	X	0	X	0
1	X	X	1	1
Z	X	0	1	Z

A	Y
X	X
0	1
1	0
Z	X

A/B	X	0	1	Z
X	X	0	X	X
0	0	0	0	0
1	X	0	1	X
Z	X	0	X	X

A/B	X	0	1	Z
X	X	X	1	X
0	X	0	1	X
1	1	1	1	1
Z	X	X	1	X

# Agenda



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

1. Einleitung

2. Algorithmische Logikminimierung

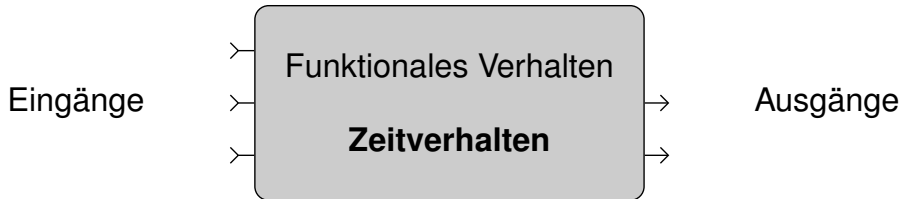
3. Mehrwertige Logik

4. Zeitverhalten

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

- ▶ Eingänge
- ▶ Ausgänge
- ▶ Spezifikation der realisierten (boole'schen) Funktion
- ▶ Spezifikation des Zeitverhaltens

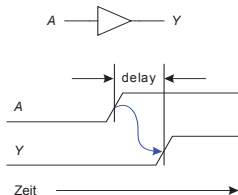




- ▶ Kombinatorisch: Werte der Ausgänge hängen nur von Werten an Eingängen ab
- ▶ reale Schaltungselemente benötigen aber endliche Zeit, um Änderung am Eingang auf Ausgang zu übertragen
  - ▶ z.B. für Umladen von MOSFET Gate-Kapazitäten

## ⇒ Zentrale Fragen

- ▶ Wann sind die Ausgänge stabil?
- ▶ Gibt es funktional äquivalente Schaltungen mit geringerer Verzögerung?
- ▶ Timing-Analyse anspruchsvoll, denn
  - ▶ Eingang kann Ausgang über verschiedene Pfade beeinflussen
  - ▶ Verzögerung kann für steigende/fallende Flanken unterschiedlich sein
  - ▶ Verzögerungen im (Sub-)Nanosekundenbereich



# Ausbreitungs- und Kontaminationsverzögerung

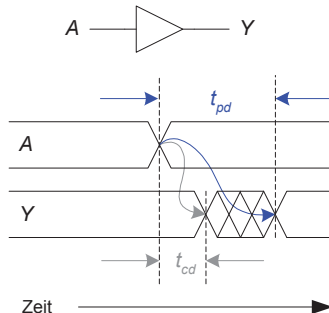
## propagation and contamination delay



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

$t_{pd}$  maximale Zeit vom Eingang zum Ausgang (**Ausbreitungsverzögerung**, propagation delay)

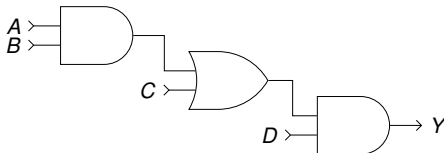
$t_{cd}$  minimale Zeit vom Eingang zum Ausgang (**Kontaminationsverzögerung**, contamination delay)





- ▶ Ursachen für Verzögerung
  - ▶ Kapazitäten, Induktivitäten und Widerstände in der Schaltung
  - ▶ Lichtgeschwindigkeit als maximale Ausbreitungsgeschwindigkeit: 30 cm/ns
- ▶ Warum können  $t_{pd}$  und  $t_{cd}$  unterschiedlich sein?
  - ▶ mehrere Ein- und Ausgänge mit unterschiedlich langen Pfaden
  - ▶ unterschiedliche Verzögerungen für steigende ( $t_{pd,LH}$ ) und fallende ( $t_{pd,HL}$ ) Flanken (Laden/Entladen des Gates in MOS Transistor)
  - ▶ Schaltungen werden
    - ▶ langsamer bei Erwärmung (Hitze erhöht Widerstand des leitfähigen Materials)
    - ▶ schneller bei Abkühlung

# Kritische (lange) und kurze Pfade



$$\begin{aligned}\text{Kritischer Pfad } t_{pd,Y} &= \max(t_{pd,AND} + t_{pd,OR} + t_{pd,AND}, \\ &\quad t_{pd,OR} + t_{pd,AND}, \\ &\quad t_{pd,AND}) \\ &= 2t_{pd,AND} + t_{pd,OR}\end{aligned}$$

$$\begin{aligned}\text{Kurzer Pfad } t_{cd,Y} &= \min(t_{cd,AND} + t_{cd,OR} + t_{cd,AND}, \\ &\quad t_{cd,OR} + t_{cd,AND}, \\ &\quad t_{cd,AND}) \\ &= t_{cd,AND}\end{aligned}$$





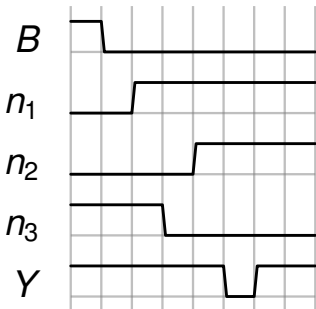
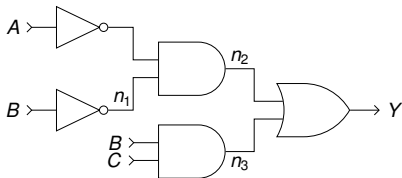
- ▶ eine Änderung eines Eingangs verursacht mehrere Änderungen des Ausgangs
- ▶ können durch geeignete Entwurfsdisziplin entschärft werden
  - ▶ Ausgänge nur zu bestimmten Zeiten auswerten (synchroner Entwurf)
  - ▶ Pfade modifizieren / hinzufügen
  - ▶ nicht alle Störimpulse können eliminiert werden (z.B. gleichzeitiges Schalten mehrerer Eingänge)
- ▶ können durch Timing- und Karnaugh-Diagramme analysiert werden

# Beispiel für Störimpuls: Erkennen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Was passiert, wenn  $(A, B, C)$  von  $(0, 1, 1)$  nach  $(0, 0, 1)$  schaltet?
- ▶  $t_{pd,NOT} = 1$ ,  $t_{pd,AND/OR} = 2$



Y:

		A			
C	AB	00	01	11	10
	0	1			
C	1	1	1	1	

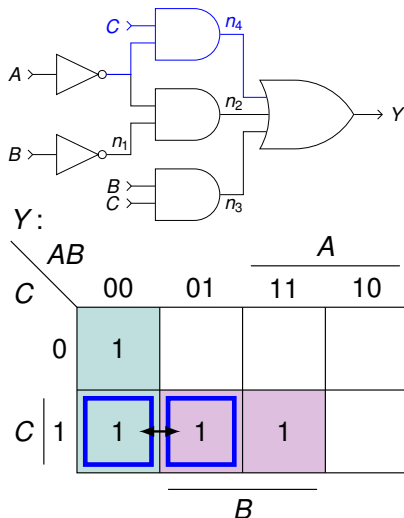
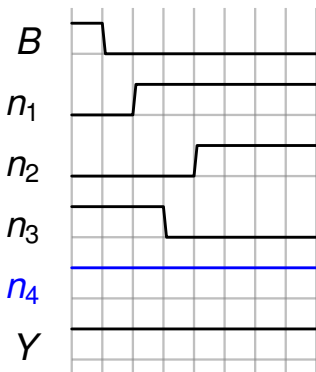
$B$

# Beispiel für Störimpuls: Beheben



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- Kritische Stelle im Karnaugh-Diagramm mit zusätzlichem Implikanten  $\bar{A} C$  überdecken



# Agenda



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT










## 1. Einleitung

## 2. Algorithmische Logikminimierung

## 3. Mehrwertige Logik

## 4. Zeitverhalten

## 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



- ▶ Kombinatorische Logik
  - ▶ Algorithmische Logikminimierung
  - ▶ Vierwertige Logik
  - ▶ Zeitverhalten
  
- ▶ Nächste Vorlesung behandelt
  - ▶ Arithmetische Grundsaltungen
  - ▶ Sequentielle Schaltungen