Digitaltechnik
Wintersemester 2017/2018
6. Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

Inhalt

TECHNISCHE UNIVERSITÄT DARMSTADT

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Rückblick auf die letzte Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

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

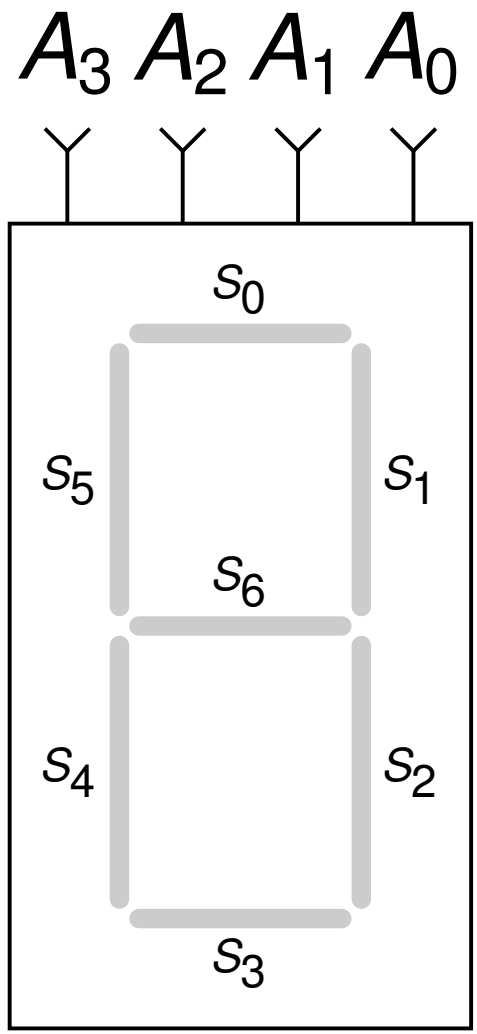
Harris 2013
Kap. 2.4,2.5,2.7,2.8

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

TECHNISCHE UNIVERSITÄT DARMSTADT

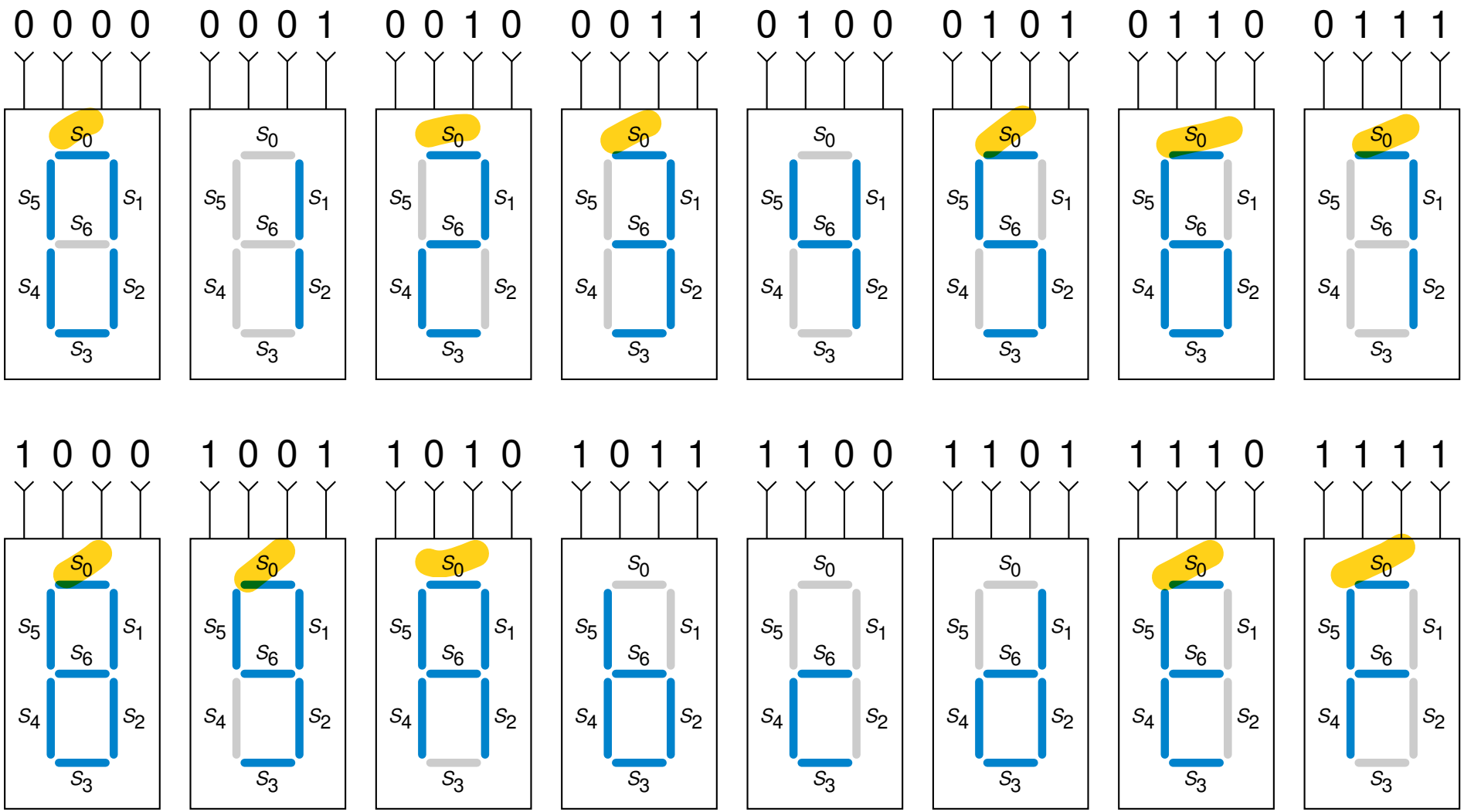
- (typ.) vier Eingänge für dargestelltes Zeichen
- sieben *unabhängig* ein-/ausschaltbare Segmente

$\Rightarrow$ jedes Segment nur für bestimmte Zeichen aktiv



Hexadezimale 7-Segment Anzeige

TECHNISCHE UNIVERSITÄT DARMSTADT



Hexadezimale 7-Segment Anzeige
Wahrheitswertetabelle

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	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

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

Hexadezimale 7-Segment Anzeige
Normalformen

TECHNISCHE UNIVERSITÄT DARMSTADT

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

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

Hexadezimale 7-Segment Anzeige
Verkürzte Minterm/Maxterm-Schreibweise

TECHNISCHE UNIVERSITÄT DARMSTADT

- Boole'sche Funktion eindeutig spezifiziert durch
  - Indizes der 1-Minterme („on set“) bzw.
  - Indizes der 0-Maxterme („off set“)

$\Rightarrow$ erlaubt kompaktere Schreibweise

Hexadezimale 7-Segment Anzeige
Verkürzte Minterm/Maxterm-Schreibweise

TECHNISCHE UNIVERSITÄT DARMSTADT

- Boole'sche Funktion eindeutig spezifiziert durch
  - Indizes der 1-Minterme („on set“) bzw.
  - Indizes der 0-Maxterme („off set“)

$\Rightarrow$ erlaubt kompaktere Schreibweise

$$S_0 = m_0 + m_2 + m_3 + m_5 + m_6 + m_7 + m_8 + m_9 + m_{10} + m_{14} + m_{15}$$

$$= \sum m(0, 2, 3, 5, 6, 7, 8, 9, 10, 14, 15)$$

$$= M_1\ M_4\ M_{11}\ M_{12}\ M_{13}$$

$$= \prod M(1, 4, 11, 13)$$

Hexadezimale 7-Segment Anzeige
Verkürzte Minterm/Maxterm-Schreibweise

TECHNISCHE UNIVERSITÄT DARMSTADT

- Boole'sche Funktion eindeutig spezifiziert durch
  - Indizes der 1-Minterme („on set“) bzw.
  - Indizes der 0-Maxterme („off set“)

$\Rightarrow$ erlaubt kompaktere Schreibweise

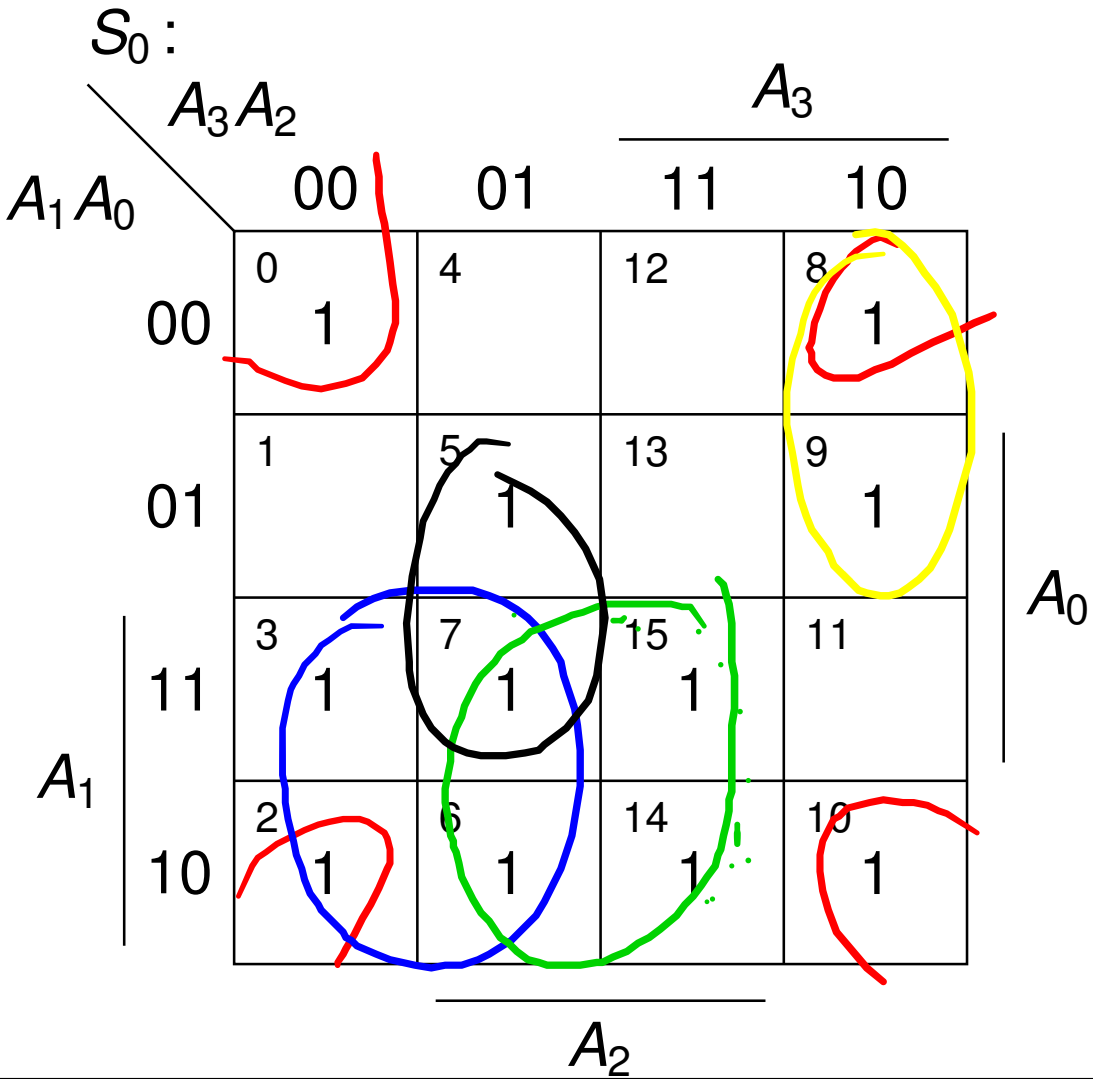
- *Achtung*: Bezug zu (Reihenfolge der) Eingangsvariablen geht verloren

$$S_0 = m_0 + m_2 + m_3 + m_5 + m_6 + m_7 + m_8 + m_9 + m_{10} + m_{14} + m_{15}$$
$$= \sum m(0, 2, 3, 5, 6, 7, 8, 9, 10, 14, 15)$$

$$= M_1\ M_4\ M_{11}\ M_{12}\ M_{13}$$
$$= \prod M(1, 4, 11, 13)$$

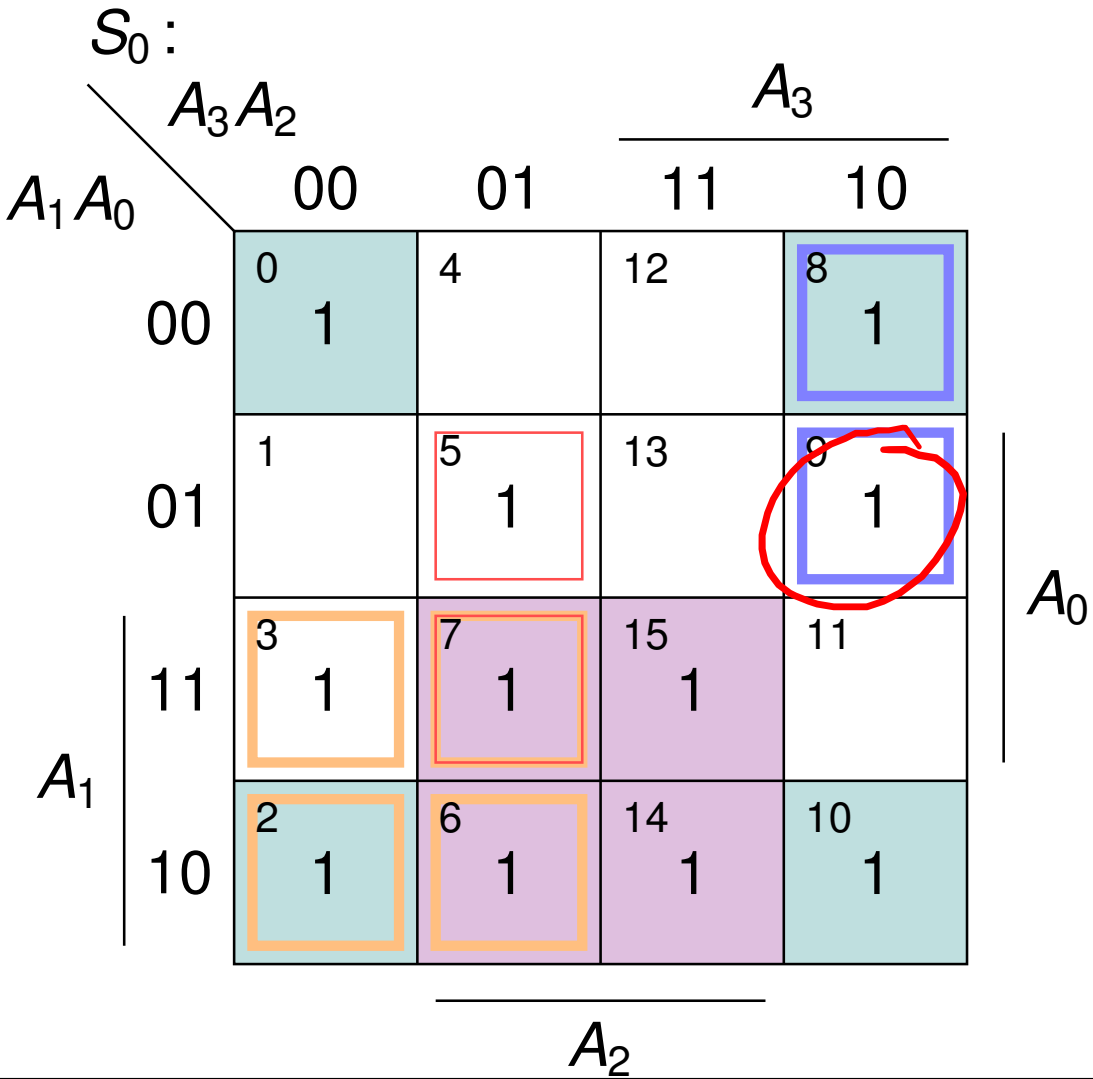
Hexadezimale 7-Segment Anzeige
Karnaugh Diagramm

TECHNISCHE UNIVERSITÄT DARMSTADT



Hexadezimale 7-Segment Anzeige
Karnaugh Diagramm

TECHNISCHE UNIVERSITÄT DARMSTADT



$$S_0 = \overline{A_0}\,\overline{A_2} + A_1\,A_2 + A_1\,\overline{A_3} + A_0\,A_2\,\overline{A_3} + \overline{A_1}\,\overline{A_2}\,A_3$$

Dezimale 7-Segment Anzeige
Wahrheitswertetabelle 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

$\Rightarrow$ in DNF und KNF gleichermaßen enthalten

$$S_0 = m_0 + m_2 + m_3 + m_5 + m_6 + m_7 + m_8 + m_9 + d_{10} + d_{11} + d_{12} + d_{13} + d_{14} + d_{15}$$

$$= \sum m(0, 2, 3, 5, 6, 7, 8, 9) + \sum d(10, 11, 12, 13, 14, 15)$$

$$= M_1 \; M_4 \; D_{10} \; D_{11} \; D_{12} \; D_{13} \; D_{14} \; D_{15}$$

$$= \prod M(1, 4) \; \prod D(10, 11, 12, 13, 14, 15)$$

Dezimale 7-Segment Anzeige
Minterm/Maxterm-Schreibweise mit Don't Cares

TECHNISCHE UNIVERSITÄT DARMSTADT

- Don't Cares können als 0 oder 1 realisiert werden

$\Rightarrow$ in DNF und KNF gleichermaßen enthalten

- *Achtung*: nur für verkürzte Schreibweise in einem Ausdruck möglich

$$S_0 = m_0 + m_2 + m_3 + m_5 + m_6 + m_7 + m_8 + m_9 + d_{10} + d_{11} + d_{12} + d_{13} + d_{14} + d_{15}$$

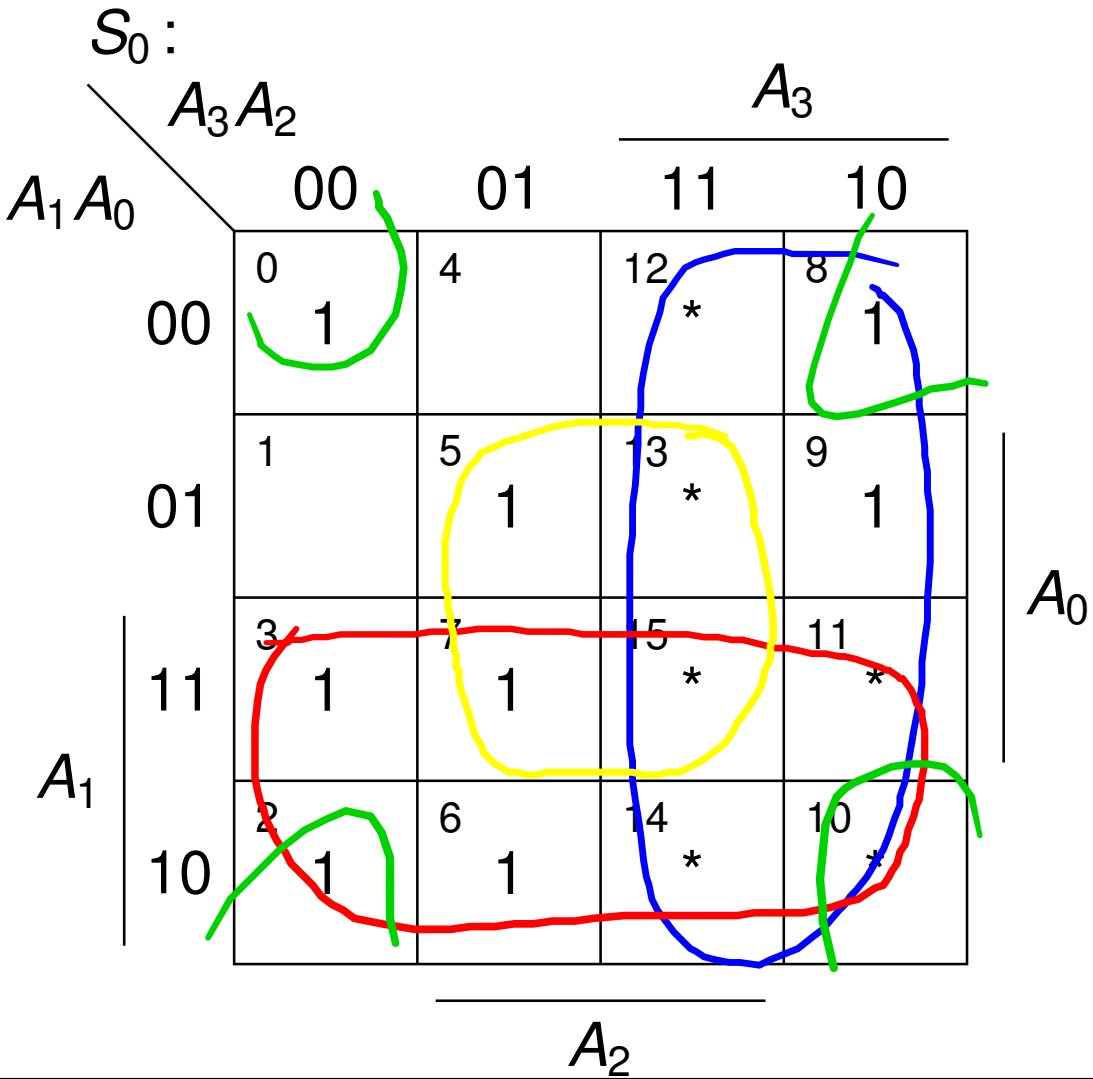
$$= \sum m(0, 2, 3, 5, 6, 7, 8, 9) + \sum d(10, 11, 12, 13, 14, 15)$$

$$= M_1 \; M_4 \; D_{10} \; D_{11} \; D_{12} \; D_{13} \; D_{14} \; D_{15}$$

$$= \prod M(1, 4) \; \prod D(10, 11, 12, 13, 14, 15)$$

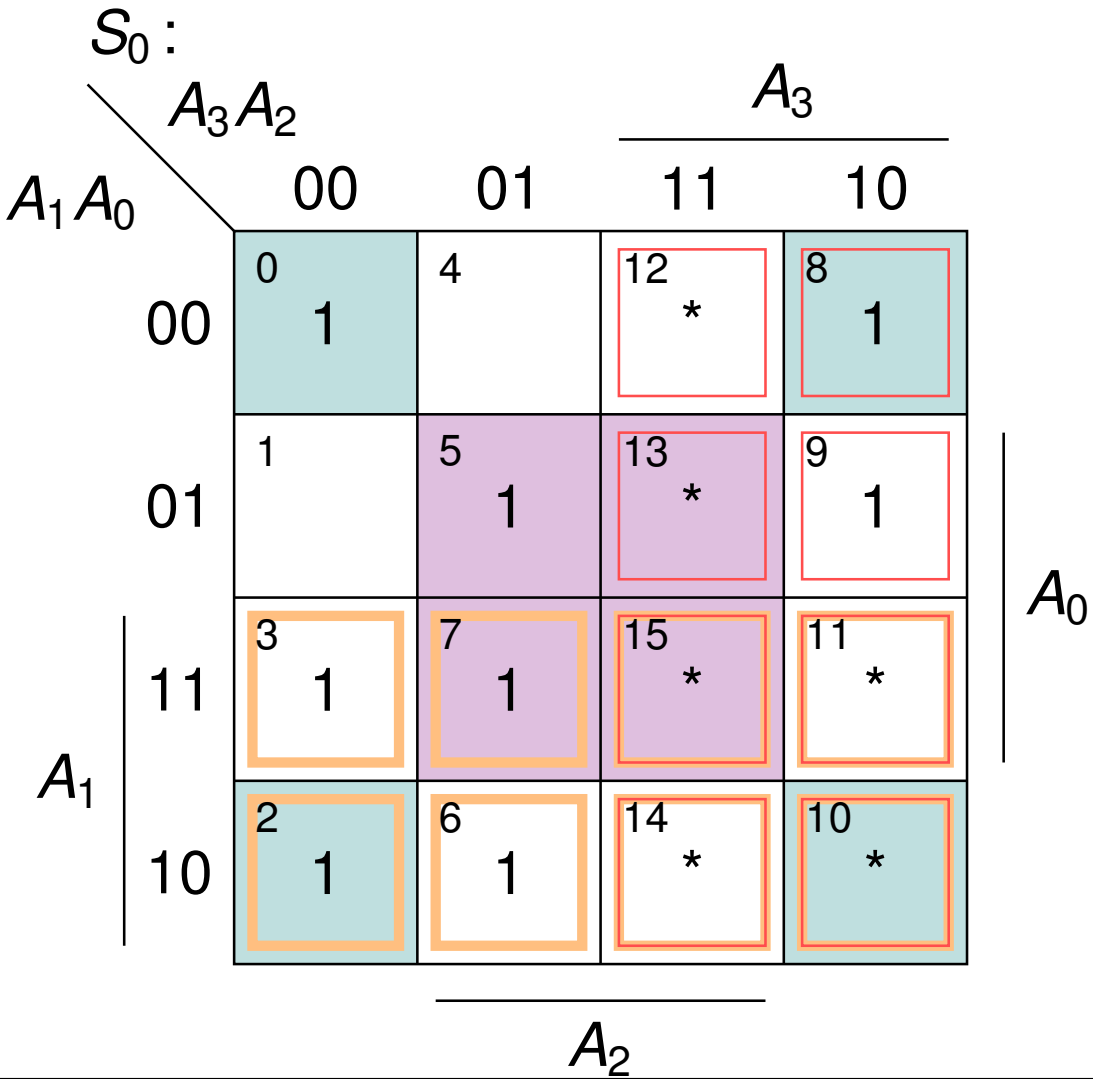
Dezimale 7-Segment Anzeige
Karnaugh Diagramm mit Don't Cares

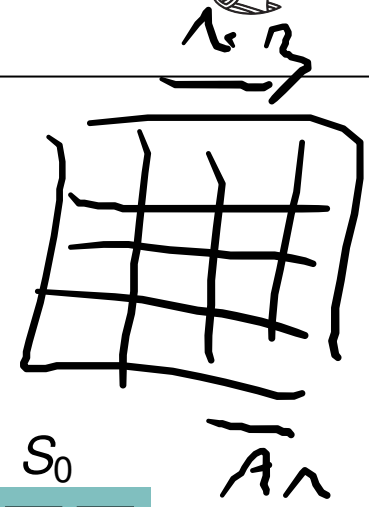
TECHNISCHE UNIVERSITÄT DARMSTADT



Dezimale 7-Segment Anzeige
Karnaugh Diagramm mit Don't Cares

TECHNISCHE UNIVERSITÄT DARMSTADT





$$S_0 = \overline{A_0}\,\overline{A_2} + A_0\,A_2 + A_1 + A_3$$

Dezimale 7-Segment Anzeige
Karnaugh Diagramm mit Maxtermen

TECHNISCHE UNIVERSITÄT DARMSTADT

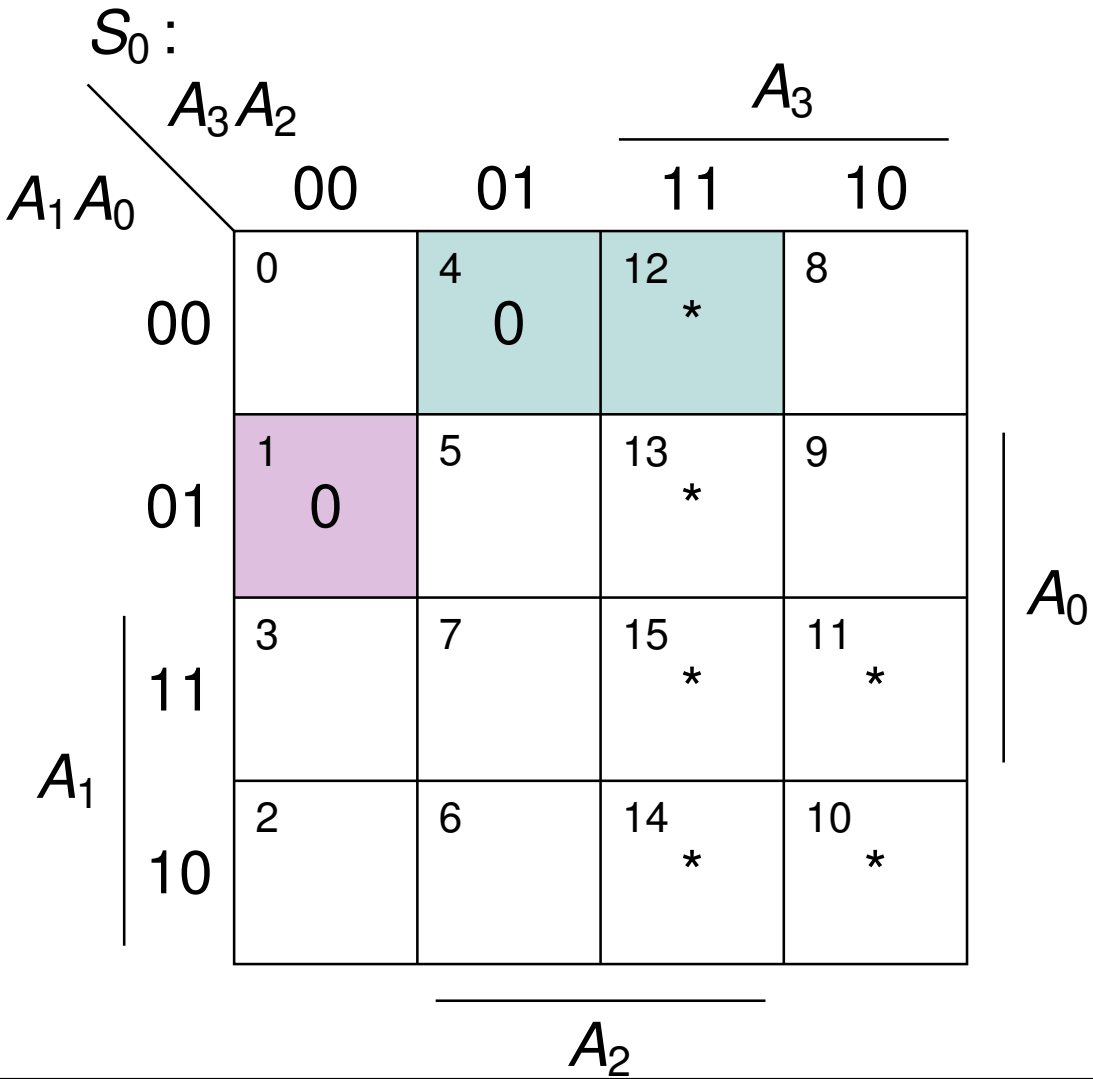
S_0 :

A_1A_0 \ A_3A_2	00	01	11	10
00	0	4 0	12 *	8
01	1 0	5	13 *	9
11	3	7	15 *	11 *
10	2	6	14 *	10 *

A_3 (über 11, 10) A_2 (unter 11, 10) A_1 (links 11, 10) A_0 (rechts 01, 11)

Dezimale 7-Segment Anzeige
Karnaugh Diagramm mit Maxtermen

TECHNISCHE UNIVERSITÄT DARMSTADT



$$\overline{S_0} = \overline{A_0}\,\overline{A_1}\,A_2 + A_0\,\overline{A_1}\,\overline{A_2}\,\overline{A_3}$$

$$S_0 = (A_0 + A_1 + \overline{A_2}) \cdot (\overline{A_0}\,A_1\,A_2\,A_3)$$

Überblick der heutigen Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Algorithmische Logikminimierung
- Vierwertige Logik
- Zeitverhalten

Harris 2013
Kap. 2.6,2.2.9

Katz 2005
Kap 3.2

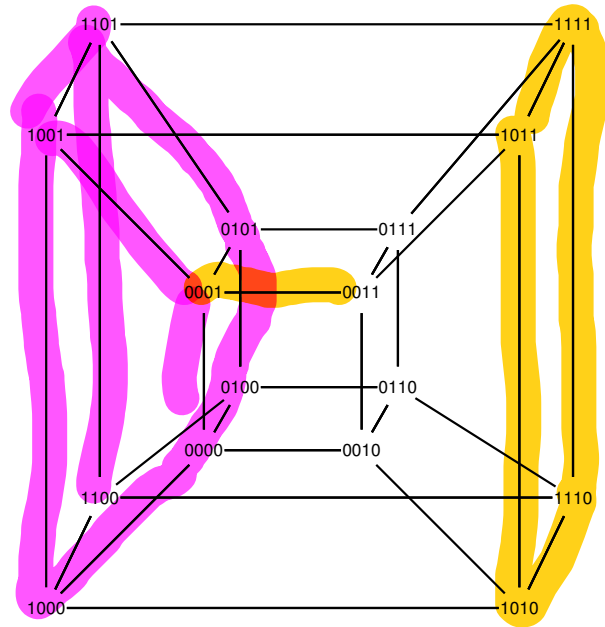
Algorithmische Logikminimierung

TECHNISCHE UNIVERSITÄT DARMSTADT

Beispiele für Verfahren zur Logikminimierung

TECHNISCHE UNIVERSITÄT DARMSTADT

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



⇒ Minimiere Anzahl der zur Darstellung einer Funktion notwendigen Implikanten

Verwendbarkeit der Verfahren

TECHNISCHE UNIVERSITÄT DARMSTADT

- Grafische Verfahren:
  - für viele (> 6) Eingänge nicht mehr praktikabel
  - keine Optimierung zwischen Ausdrücken für mehrere Ausgänge

Verwendbarkeit der Verfahren

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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

  $\Rightarrow$ Rechzeit steigt exponentiell mit Anzahl der Eingänge

Verwendbarkeit der Verfahren

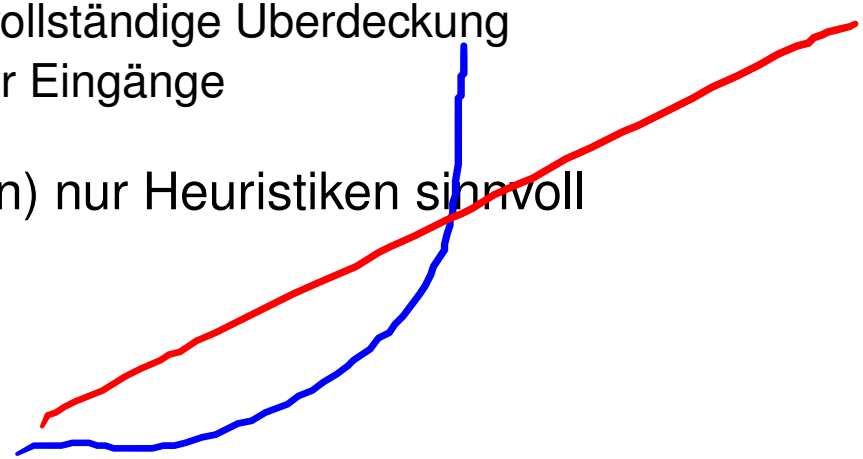
TECHNISCHE UNIVERSITÄT DARMSTADT

- 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

  $\Rightarrow$ Rechzeit steigt exponentiell mit Anzahl der Eingänge

$\Rightarrow$ für wirklich große Probleme (> 50 Variablen) nur Heuristiken sinnvoll

- geringere Laufzeitkomplexität
- geringere Lösungsqualität



Espresso-Heuristik

TECHNISCHE UNIVERSITÄT DARMSTADT

- in 1980er Jahren bei IBM und UC Berkeley entwickel
- unterstützt auch mehrere (zusammen optimierte) Ausgänge
- Details des Algorithmus hier nicht relevant (vgl. Katz 2005 bzw. Rudell 1986)
- hier nur Anwendung einer konkreten Implementierung

Espresso-Heuristik

TECHNISCHE UNIVERSITÄT DARMSTADT

- in 1980er Jahren bei IBM und UC Berkeley entwickel
- unterstützt auch mehrere (zusammen optimierte) Ausgänge
- Details des Algorithmus hier nicht relevant (vgl. Katz 2005 bzw. Rudell 1986)
- hier nur Anwendung einer konkreten Implementierung
  - `https://embedded.eecs.berkeley.edu/pubs/downloads/espresso`
  - Anleitung / Quellen auch im 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
```

Espresso Dateiformat

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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` Dateiende

Espresso Dateiformat

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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` Dateiende
- Wahrheitswertetabelle im ASCII Format
  - 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
    - `1` Implikant im on set des Ausgangs
    - `-` Implikant im on set *oder* off set des Ausgangs (Don't Care)

Espresso Dateiformat

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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` Dateiende
- Wahrheitswertetabelle im ASCII Format
  - 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
    - `1` Implikant im on set des Ausgangs
    - `-` Implikant im on set *oder* off set des Ausgangs (Don't Care)
- „#“ leitet Kommentar ein

Espresso Minimalbeispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

xor.esp

```
1  # Espresso description of Y = A xor B
2  .i    2
3  .o    1
4  .ilb  A  B   # optional
5  .ob   Y      # optional
6  .p    4      # optional
7  00  0        # optional
8  01  1
9  10  1
10 11  0        # optional
11 .e           # optional
```

Espresso 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  -
```

Espresso 7-Segment Anzeige
Eingabedateien

TECHNISCHE UNIVERSITÄT DARMSTADT

sevenseg/s0.esp

```
# S0 of 7-segment display
.i    4
.o    1
0000  1
0010  1
0011  1
0101  1
0110  1
0111  1
1000  1
1001  1
1010  -
1011  -
1100  -
1101  -
1110  -
1111  -
```

sevenseg/all.esp

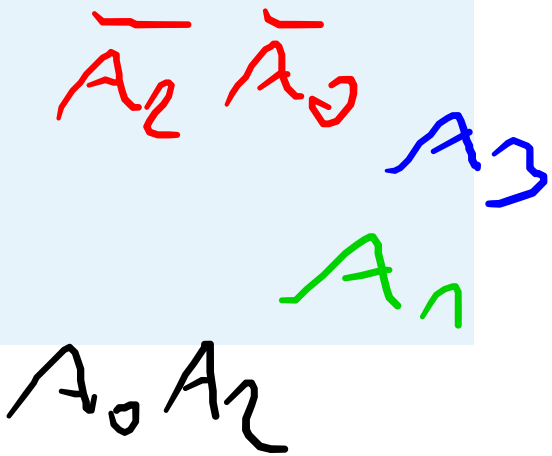
```
# 7-segment display
.i    4
.o    7
0000  1111110
0001  0110000
0010  1101101
0011  1111001
0100  0110011
0101  1011011
0110  1011111
0111  1110000
1000  1111111
1001  1111011
1010  -------
1011  -------
1100  -------
1101  -------
1110  -------
1111  -------
```

Espresso 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
```



Espresso 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
```

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
```

Espresso kann noch viel mehr

TECHNISCHE UNIVERSITÄT DARMSTADT

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

Mehrwertige Logik

TECHNISCHE UNIVERSITÄT DARMSTADT

Mehrwertige Logik

TECHNISCHE UNIVERSITÄT DARMSTADT

- bisher galt:
  - jeder Schaltungsknoten (außer Eingänge) wird von *genau einem* Schaltungselement auf 0 oder 1 getrieben
  - Axiome der boole'schen Algebra basieren auf $\mathbb{B} = \{0, 1\}$

$\Rightarrow$ ignoriert wichtige Teile der Realität

- Wie breiten sich ungültige Spannungen in Schaltung aus?
- Können ungültige Spannungsbereiche gezielt eingesetzt werden?

Mehrwertige Logik

TECHNISCHE UNIVERSITÄT DARMSTADT

- bisher galt:
  - jeder Schaltungsknoten (außer Eingänge) wird von *genau einem* Schaltungselement auf 0 oder 1 getrieben
  - Axiome der boole'schen Algebra basieren auf $\mathbb{B} = \{0, 1\}$

$\Rightarrow$ ignoriert wichtige Teile der Realität
  - Wie breiten sich ungültige Spannungen in Schaltung aus?
  - Können ungültige Spannungsbereiche gezielt eingesetzt werden?

$\Rightarrow$ Unterscheidung von zwei weiteren Logikwerten zwischen 0 und 1
  - X mehrfach getrieben (fehlerhaft)
  - Z ungetrieben (gezielt)

Mehrwertige Logik

TECHNISCHE UNIVERSITÄT DARMSTADT

- bisher galt:
  - jeder Schaltungsknoten (außer Eingänge) wird von *genau einem* Schaltungselement auf 0 oder 1 getrieben
  - Axiome der boole'schen Algebra basieren auf $\mathbb{B} = \{0, 1\}$

$\Rightarrow$ ignoriert wichtige Teile der Realität
  - Wie breiten sich ungültige Spannungen in Schaltung aus?
  - Können ungültige Spannungsbereiche gezielt eingesetzt werden?

$\Rightarrow$ Unterscheidung von zwei weiteren Logikwerten zwischen 0 und 1
  - X mehrfach getrieben (fehlerhaft)
  - Z ungetrieben (gezielt)

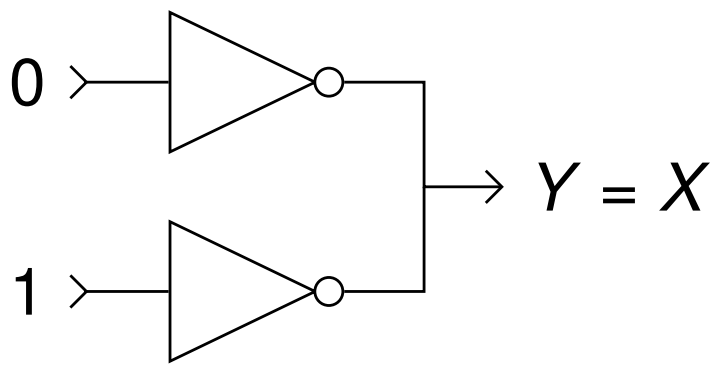
- *Achtung:*
  - nicht mit „Don't Care" (*) verwechseln
  - tatsächliche Spannung *kann* auch im 0- oder 1-Bereich liegen, das Schaltungsdesign stellt dies aber nicht sicher

Konkurrierende Ausgänge: X

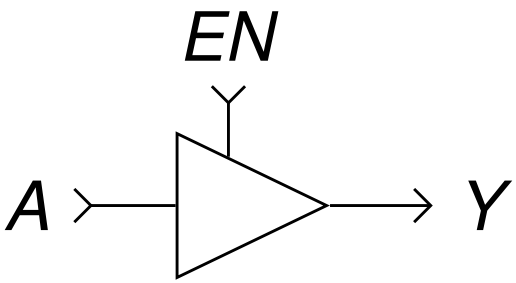
TECHNISCHE UNIVERSITÄT DARMSTADT

- mehrere (unabhängige) Treiber für den selben Schaltungsknoten
- Konflikt, sobald Treiber in entgegengesezte Richtung ziehen
  - instabil: abhängig von Betriebsspannung, Temperatur, etc.
  - destruktiv: Kurzschluss verursacht hohen Energieverbrauch
- fast immer ein Entwurfsfehler
  - bspw. doppelte Zuweisung in Hardwarebeschreibung

  $\Rightarrow$ Konflikt-Quelle muss in Simulation leicht nachvollziehbar sein



Tristate-Buffer: Z



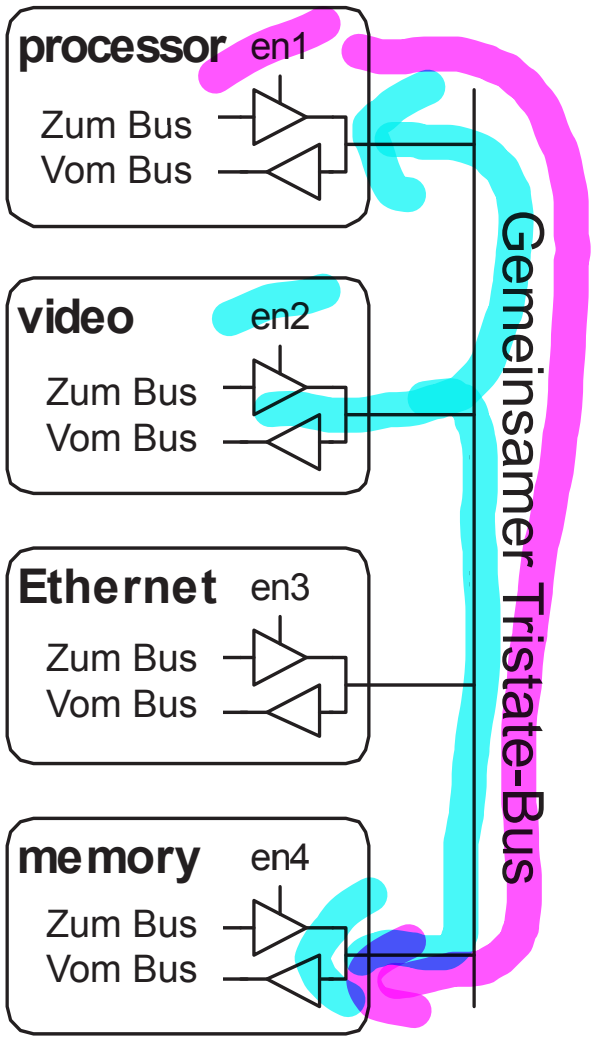
EN	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

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

Tristate-Buffer für Busse

TECHNISCHE UNIVERSITÄT DARMSTADT

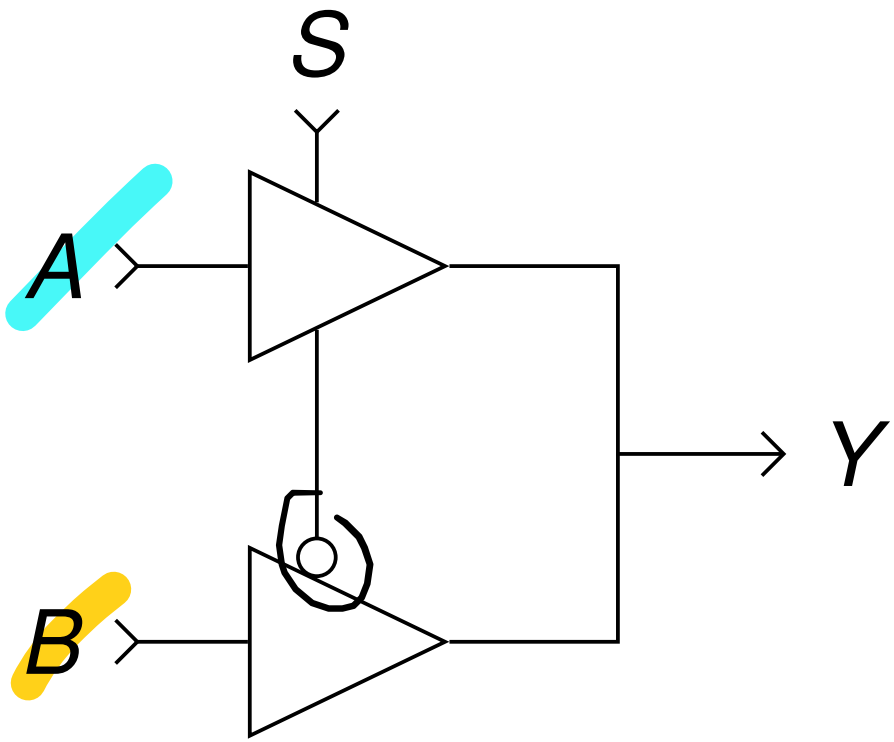
- mehrere Treiber an gemeinsamer Leitung
- zu jedem Zeitpunkt *genau ein* aktiver Treiber
- erlaubt Wechsel der Kommunikationsrichtung



Tristate-Buffer für

TECHNISCHE UNIVERSITÄT DARMSTADT

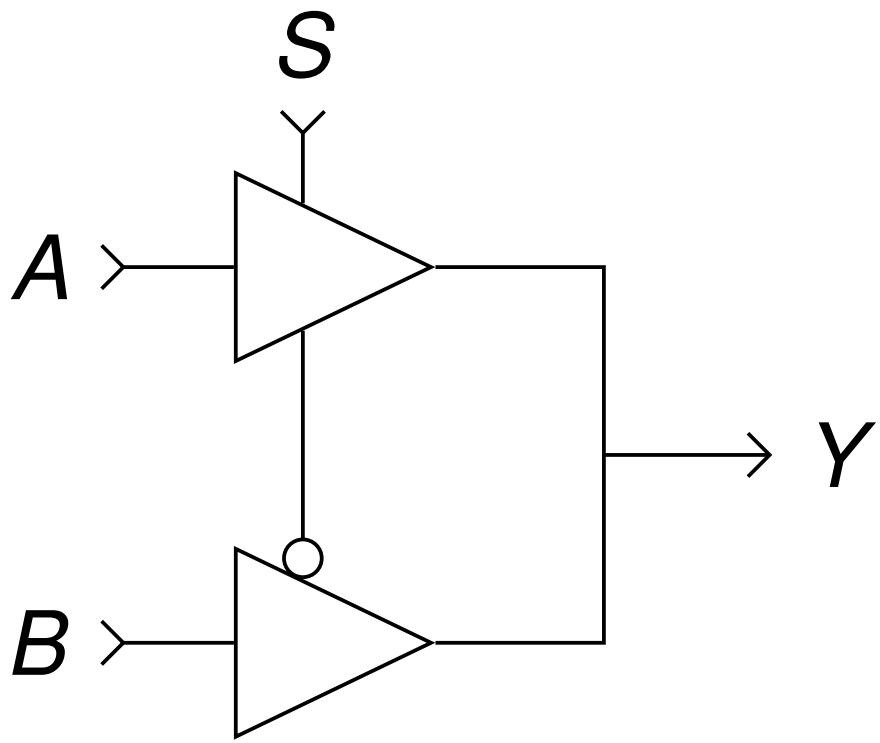
S	A	B	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Tristate-Buffer für Multiplexer

TECHNISCHE UNIVERSITÄT DARMSTADT

S	A	B	Y
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

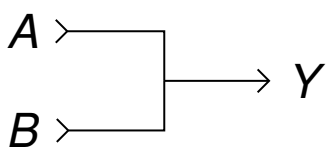


Mehrwertige Logik in Schaltnetzen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Resolutionstabellen definieren Ausbreitung von X und Z
- mehr Konvention (für Simulator) als physikalische Realität
- bspw. 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



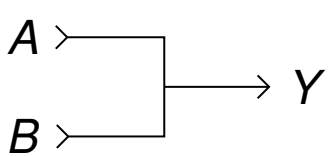
Mehrwertige Logik in Schaltnetzen

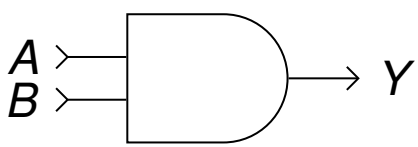
TECHNISCHE UNIVERSITÄT DARMSTADT

- Resolutionstabellen definieren Ausbreitung von X und Z
- mehr Konvention (für Simulator) als physikalische Realität
- bspw. 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/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





Mehrwertige Logik in Schaltnetzen

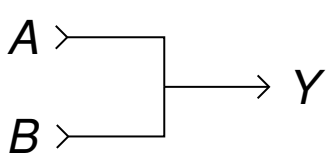
TECHNISCHE UNIVERSITÄT DARMSTADT

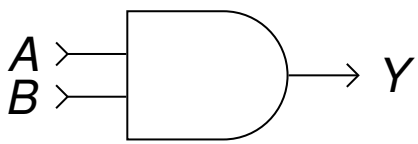
- Resolutionstabellen definieren Ausbreitung von X und Z
- mehr Konvention (für Simulator) als physikalische Realität
- bspw. IEEE 1164:

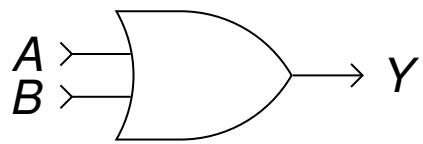
A/B	X	0	1	Z
X	X	X	X	X
0	X	0	X	0
1	X	X	1	1
Z	X	0	1	Z

A/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







Mehrwertige Logik in Schaltnetzen

TECHNISCHE UNIVERSITÄT DARMSTADT

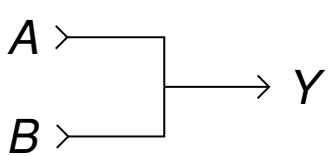
- Resolutionstabellen definieren Ausbreitung von X und Z
- mehr Konvention (für Simulator) als physikalische Realität
- bspw. IEEE 1164:

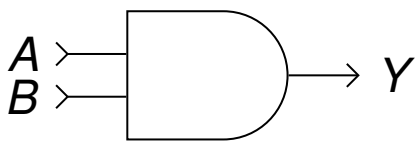
A/B	X	0	1	Z
X	X	X	X	X
0	X	0	X	0
1	X	X	1	1
Z	X	0	1	Z

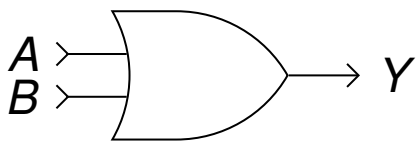
A/B	X	0	1	Z
X	X	0	X	X
0	0	0	0	0
1	X	0	1	X
Z	X	0	X	X

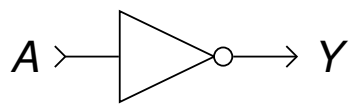
A/B	X	0	1	Z
X	X	X	1	X
0	X	0	1	X
1	1	1	1	1
Z	X	X	1	X

A	Y
X	X
0	1
1	0
Z	X









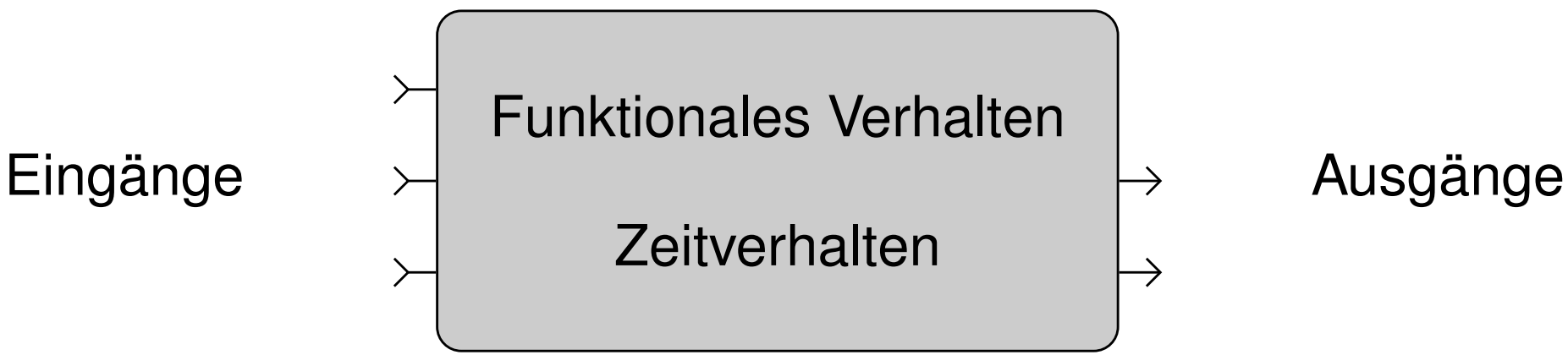
Zeitverhalten

TECHNISCHE UNIVERSITÄT DARMSTADT

Abstrakte Eigenschaften *logischer* Schaltungen

TECHNISCHE UNIVERSITÄT DARMSTADT

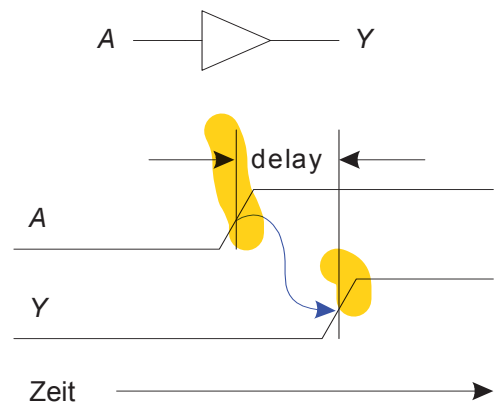
- Eingängen
- Ausgängen
- Spezifikation der realisierten (boolschen) Funktion
- Spezifikation des Zeitverhaltens



Zeitverhalten einer kombinatorischen Schaltung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Werte der Ausgänge hängen nur von Werten an Eingängen ab
- reale Schaltungselemente benötigen aber endliche Zeit, um Änderung am Eingang auf Ausgang zu übertragen
  - bspw. für Umladen von CMOS Gate-Kapazitäten



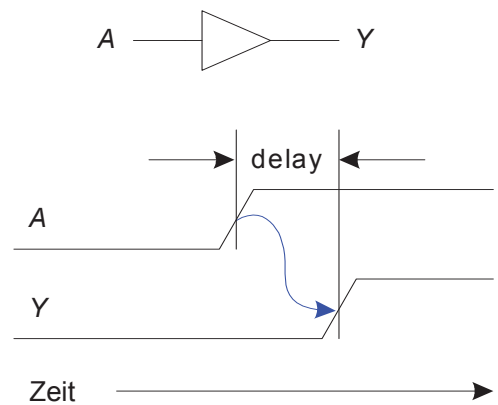
Zeitverhalten einer kombinatorischen Schaltung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Werte der Ausgänge hängen nur von Werten an Eingängen ab
- reale Schaltungselemente benötigen aber endliche Zeit, um Änderung am Eingang auf Ausgang zu übertragen
  - bspw. für Umladen von CMOS Gate-Kapazitäten

$\Rightarrow$ Zentrale Fragen

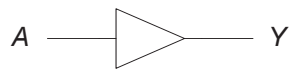
- Gibt es funktional äquivalente Schaltungen mit geringerer Verzögerung?
- Wann sind die Ausgänge stabil?



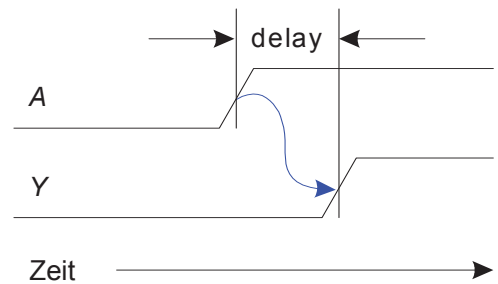
Zeitverhalten einer kombinatorischen Schaltung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Werte der Ausgänge hängen nur von Werten an Eingängen ab
- reale Schaltungselemente benötigen aber endliche Zeit, um Änderung am Eingang auf Ausgang zu übertragen
  - bspw. für Umladen von CMOS Gate-Kapazitäten



$\Rightarrow$ Zentrale Fragen
  - Gibt es funktional äquivalente Schaltungen mit geringerer Verzögerung?
  - Wann sind die Ausgänge stabil?



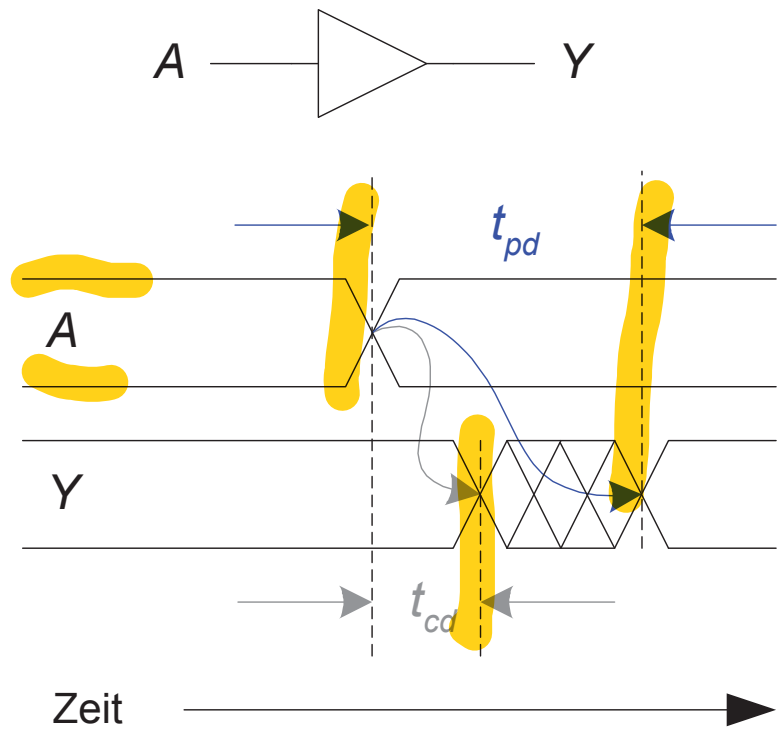
- Timing-Analyse anspruchsvoll, denn
  - Eingang kann Ausgang über verschiedene Pfade beeinflussen
  - Verzögerung kann für steigende/fallende Flanken unterschiedlich sein
  - Verzögerungen im (Sub-)Nanosekundenbereich

Ausbreitungs- und Kontaminationsverzögerung
propagation and contamination delay

TECHNISCHE UNIVERSITÄT DARMSTADT

t_{pd} maximale Zeit vom Eingang zum Ausgang (**Ausbreitungsverzögerung**)

t_{cd} minimale Zeit vom Eingang zum Ausgang (**Kontaminationsverzögerung**)



Ausbreitungs- und Kontaminationsverzögerung

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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?
  - unterschiedliche Verzögerungen für steigende ($t_{pd,LH}$) und fallende ($t_{pd,HL}$) Flanken
  - mehrere Ein- und Ausgänge mit unterschiedlich langen Pfaden
  - Schaltungen werden
    - ... langsamer bei Erwärmung
    - ... schneller bei Abkühlung

Beispiele aus der Praxis: CMOS Inverter

TECHNISCHE UNIVERSITÄT DARMSTADT

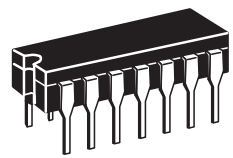
MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

MC14007UB

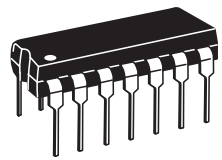
Dual Complementary Pair Plus Inverter

The MC14007UB multi–purpose device consists of three N–channel and three P–channel enhancement mode devices packaged to provide access to each device. These versatile parts are useful in inverter circuits, pulse–shapers, linear amplifiers, high input impedance amplifiers, threshold detectors, transmission gating, and functional gating.

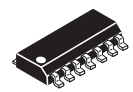
- Diode Protection on All Inputs
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low–power TTL Loads or One Low–power Schottky TTL Load Over the Rated Temperature Range
- Pin–for–Pin Replacement for CD4007A or CD4007UB
- This device has 2 outputs without ESD Protection. Anti–static precautions must be taken.



L SUFFIX
CERAMIC
CASE 632



P SUFFIX
PLASTIC
CASE 646

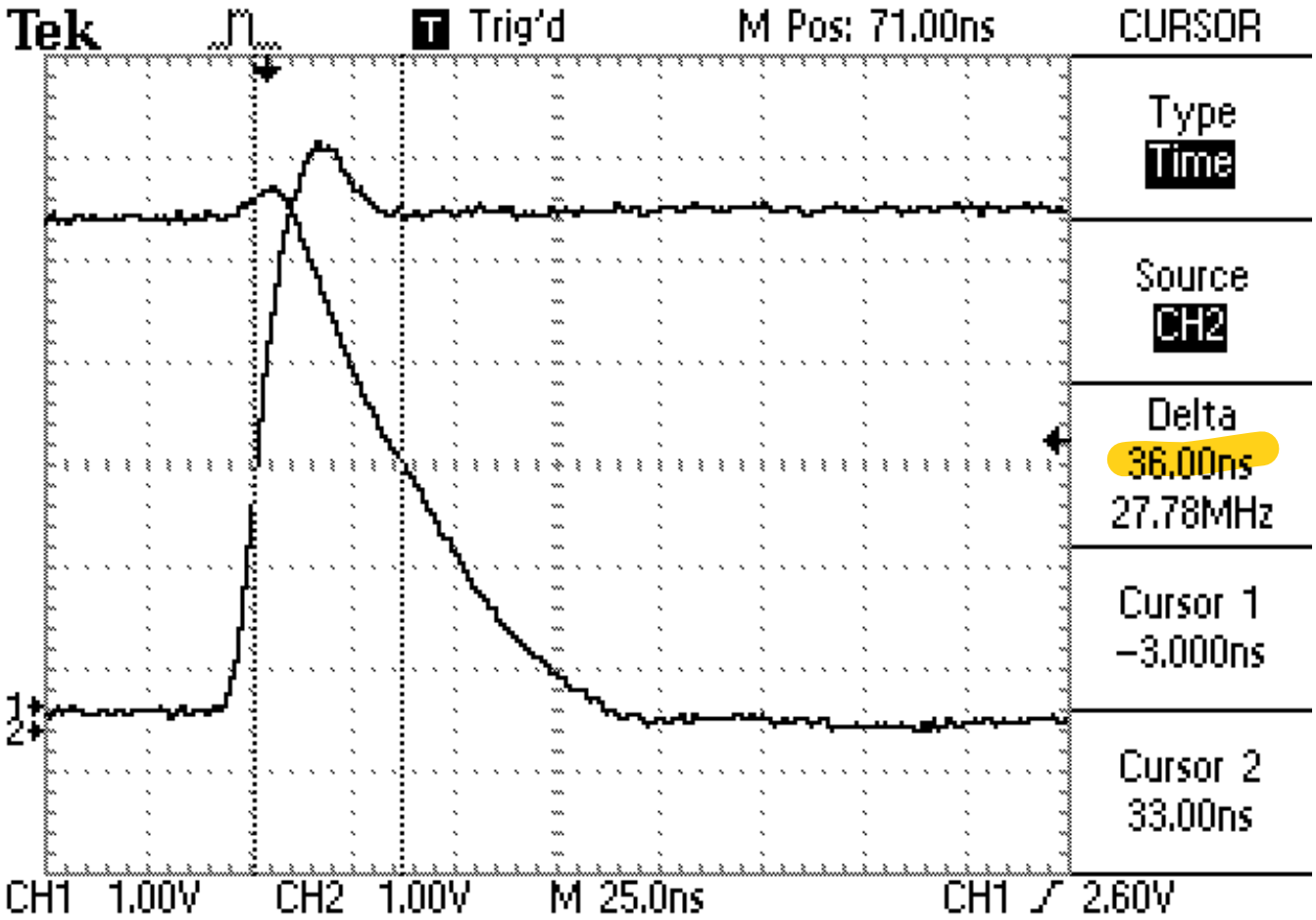


D SUFFIX
SOIC
CASE 751A

ORDERING INFORMATION

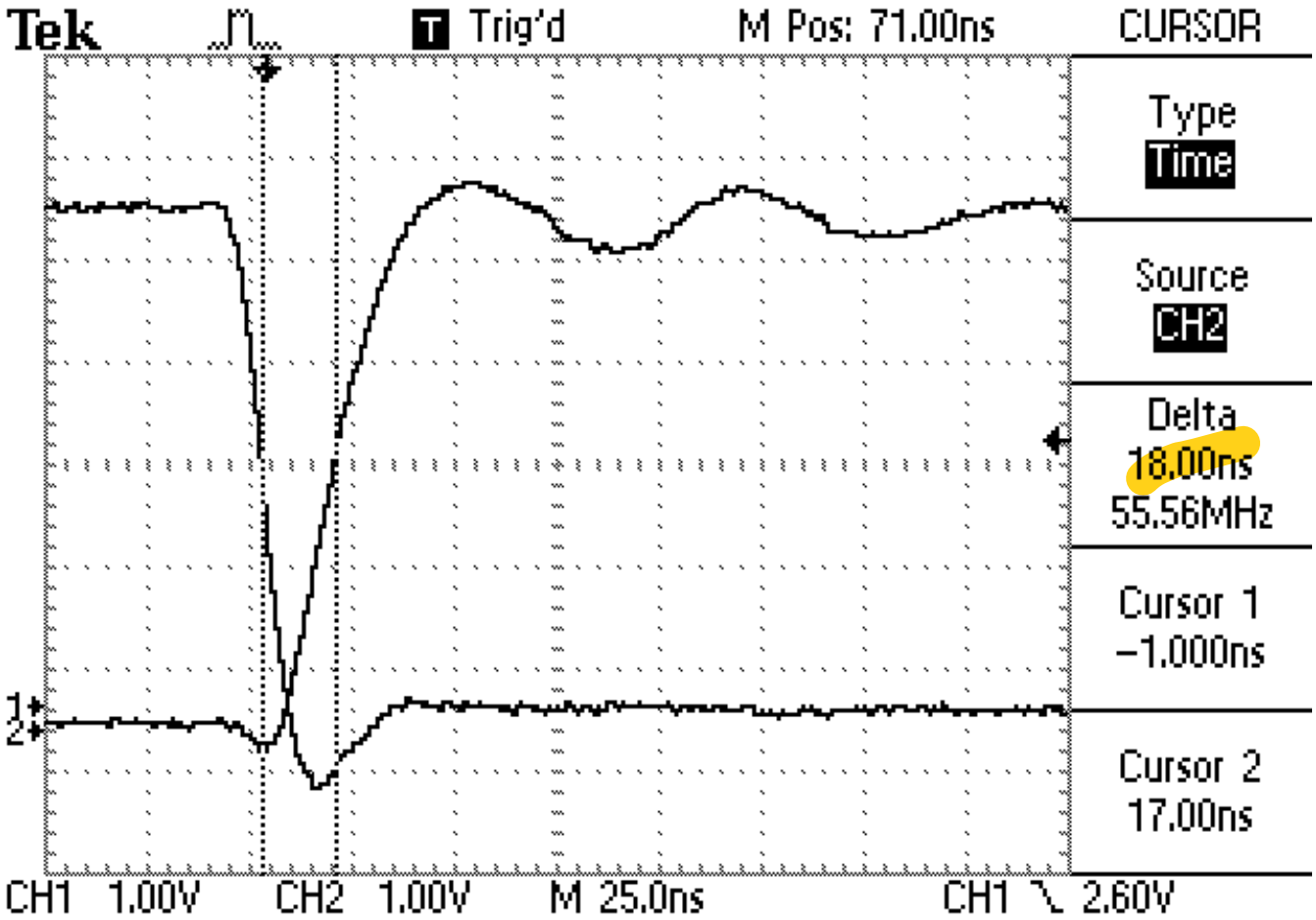
Beispiele aus der Praxis: $t_{pd,HL} \approx 36\,ns$

TECHNISCHE UNIVERSITÄT DARMSTADT

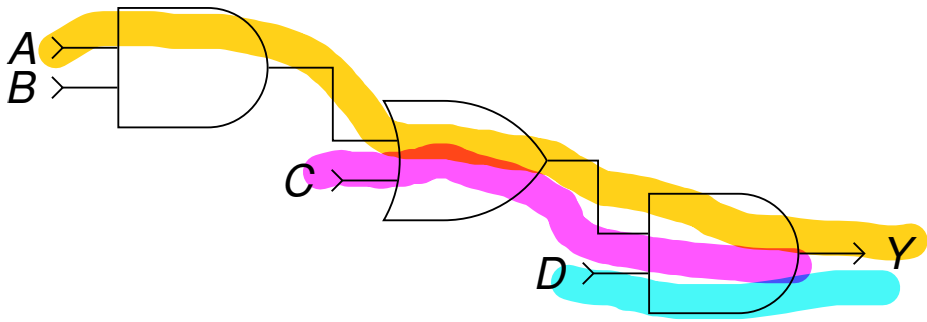


Beispiele aus der Praxis: $t_{pd,LH} \approx 18\,ns$

TECHNISCHE UNIVERSITÄT DARMSTADT



Kritische (lange) und kurze Pfade



$$t_{pd,Y} = \max(t_{pd,AND} + t_{pd,OR} + t_{pd,AND},\ t_{pd,OR} + t_{pd,AND},\ t_{pd,AND})$$

$$= 2t_{pd,AND} + t_{pd,OR}$$

Kritischer Pfad

$$t_{cd,Y} = \min(t_{cd,AND} + t_{cd,OR} + t_{cd,AND},\ t_{cd,OR} + t_{cd,AND},\ t_{cd,AND})$$

$$= t_{cd,AND}$$

Kurzer Pfad

Störimpulse (Glitches)

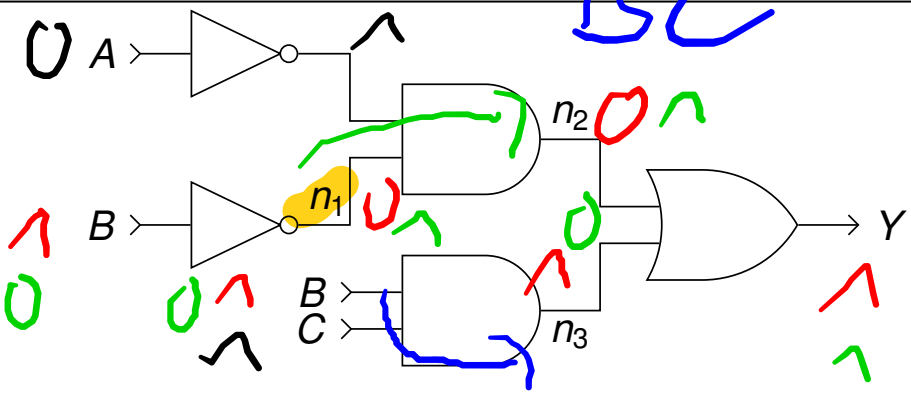
TECHNISCHE UNIVERSITÄT DARMSTADT

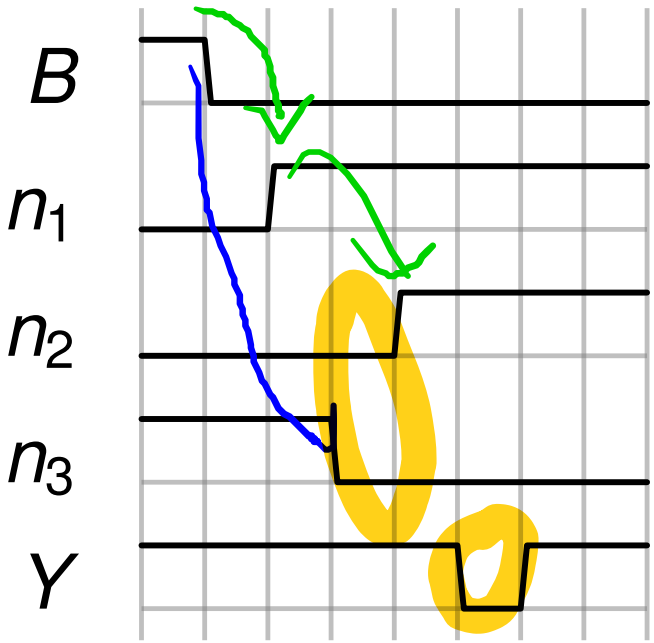
- eine Änderung eines Eingangs verursacht mehrere Änderungen des Ausgangs
- können durch geeignete Entwurfsdisziplin entschärft werden
  - Ausgänge nur zu bestimmten Zeiten auswerten (synchroner Entwurf)
  - Pfade modifizieren / hinzufügen
  - nicht alle Störimpulse können eliminiert werden
    (bspw. 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?





Y:

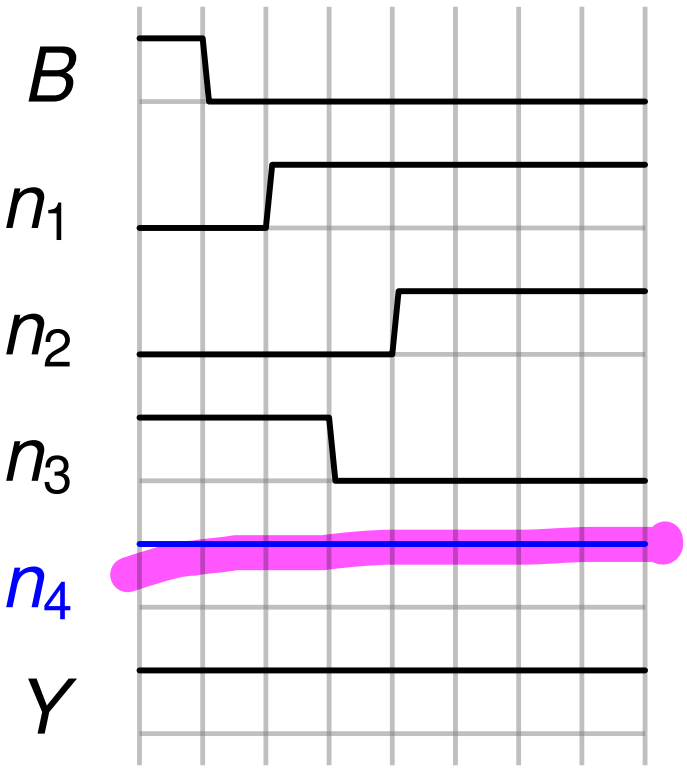
C \ AB	00	01	11	10
0	1			
1	1	1	1	

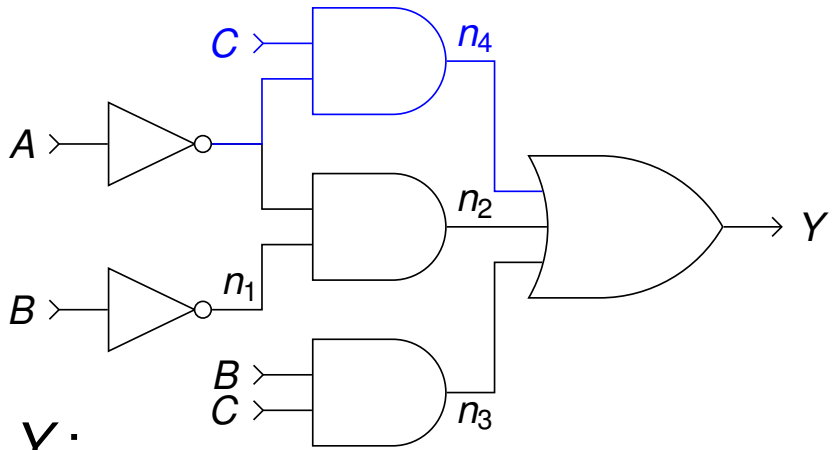
(A über 11 und 10; B unter 01 und 11)

Beispiel für Störimpuls: Beheben

TECHNISCHE UNIVERSITÄT DARMSTADT

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





Y:

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

C (rows), B (columns)

Zusammenfassung

TECHNISCHE UNIVERSITÄT DARMSTADT

Zusammenfassung und Ausblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Kombinatorische Logik
  - Algorithmische Logikminimierung
  - Vierwertige Logik
  - Zeitverhalten

- Nächste Vorlesung behandelt
  - Sequentielle Schaltungen