

Elektronikpraktikum im WS 2010/11

UNIVERSITÄT STUTTGART

Protokoll zum Versuch

## V09: Logische Gatter

Stephan Ludwig, Nicolai Lang

2. Januar 2011

### Zusammenfassung

Der folgende Versuch befasst sich mit der Funktionsweise von NAND-Gattern in TTL-Bauweise. Im ersten Teil werden die Schaltschwellen des verwendeten TTL-NAND-Gatters untersucht. Der zweite Teil befasst sich mit der Realisierung unterschiedlicher logischer Funktionen auf Basis von NAND-Gattern. Speziell werden ein Multiplexer und ein Binär-Dekoder aufgebaut.

<b>Verfasser</b> .....	Lang, Nicolai (2439501, B.Sc. Physik)
<b>Mitarbeiter</b> .....	Ludwig, Stephan (2526605, B.Sc. Physik)
<b>Gruppennummer</b> .....	1-16
<b>Versuchstag</b> .....	20. Dezember 2010
<b>Betreuer</b> .....	Torsten Randler

## Inhaltsverzeichnis

<b>1. Grundlagen</b>	<b>3</b>
1.1. Logik . . . . .	3
1.1.1. Junktoren . . . . .	3
1.1.2. Elementare Identitäten . . . . .	4
1.1.3. Elektrotechnische Realisierung . . . . .	4
1.2. Zahlensysteme . . . . .	5
1.3. Beispielrechnungen . . . . .	5
<b>2. Messprinzip, Schaltpläne und Versuchsablauf</b>	<b>7</b>
2.1. Eigenschaften von TTL-Logikbausteinen . . . . .	7
2.2. NAND als elementares Gatter . . . . .	7
2.3. Multiplexer . . . . .	8
2.4. Dekodierschaltungen . . . . .	8
<b>3. Berechnungen &amp; Formeln</b>	<b>9</b>
<b>4. Auswertung</b>	<b>10</b>
4.1. Eigenschaften von TTL-Logikbausteinen . . . . .	10
4.1.1. Funktionsweise eines TTL-Gatters DM7400 . . . . .	10
4.1.2. Schaltschwellen des TTL-Gatters DM7400 . . . . .	10
4.2. NAND als elementares Gatter . . . . .	12
4.2.1. OR-Funktion aus NAND-Gattern . . . . .	12
4.2.2. XOR-Funktion aus NAND-Gattern . . . . .	12
4.3. Multiplexer . . . . .	13
4.4. Dekodierschaltungen . . . . .	14
<b>5. Fehlerrechnung</b>	<b>16</b>
<b>6. Zusammenfassung</b>	<b>17</b>
<b>A. Messwerte</b>	<b>18</b>

# 1. Grundlagen

## 1.1. Logik

### 1.1.1. Junktoren

Die *Schaltalgebra* ist in mathematischer Hinsicht isomorph zur zweielementigen booleschen Algebra. Daher unterscheidet sie sich ausschließlich in den Bezeichnungen und Schreibweisen von der klassischen Aussagenlogik. Diese beschreibt eine algebraische Struktur, die durch ihre Trägermenge  $A$  und die auf dieser Menge definierten Funktionen ( $f_i$ ) gegeben ist. Im Folgenden behandeln wir nur die zweielementige boolesche Algebra der klassischen Aussagenlogik mit  $A = \{w, f\}$  oder  $A = \{1, 0\}$  bzw. die Schaltalgebra mit  $A = \{H, L\}$ . Die Funktionen auf dieser Trägermenge werden in der klassischen Aussagenlogik als *Junktoren* bezeichnet. Eine Menge  $\{f_i\}$  von Junktoren heißt *funktional vollständig* oder auch *Junktorbasis* wenn alle anderen Junktoren der algebraischen Struktur durch diese ausgedrückt werden können. Den Junktoren einer einelementigen Junktorbasis bezeichnet man auch als SHEFFER-Operator. Häufig benutzte Junktorbasisen in der Aussagenlogik sind  $\{\wedge, \neg\}$  oder  $\{\vee, \neg\}$ . Die häufig verwendete Junktormenge  $\{\vee, \wedge, \neg\}$  ist damit offensichtlich vollständig aber keine Basis im klassischen Sinne, da sich je einer der Operatoren  $\vee$  und  $\wedge$  durch die beiden übrigen ausdrücken lässt. Die klassische Aussagenlogik kennt nur zwei SHEFFER-Operatoren: Den *Schefferstrich*  $\uparrow$  (NAND) und den *Peirce-Operator*  $\downarrow$  (NOR). Die Tatsache, dass die NAND-Funktion funktional vollständig ist, hat weitreichende Konsequenzen für die elektrotechnische Realisierung von logischen Funktionen. In Tabelle 1 sind die wichtigsten Junktoren der Schaltalgebra mit ihren Schaltsymbolen und ihrer Definition über den Schefferstrich gegeben.

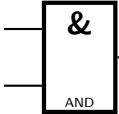
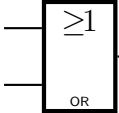
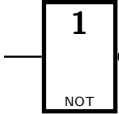
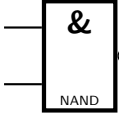
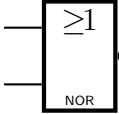
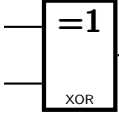
Bezeichnung	Junktor	SdSA <sup>1</sup>	SdL <sup>2</sup>	Definition via $\uparrow$	Schaltsymbol
Konjunktion	AND	$A \cdot B$	$A \wedge B$	$(A \uparrow B) \uparrow (A \uparrow B)$	
Disjunktion	OR	$A + B$	$A \vee B$	$(A \uparrow A) \uparrow (B \uparrow B)$	
Negation	NOT	$\overline{A}$	$\neg A$	$(A \uparrow A)$	
Sheffer-Funktion	NAND	$\overline{A \cdot B}$	$A \uparrow B$	Definition über die Wahrheitstafel.	
Peirce-Funktion	NOR	$\overline{A + B}$	$A \downarrow B$	$((A \uparrow A) \uparrow (B \uparrow B)) \uparrow ((A \uparrow A) \uparrow (B \uparrow B))$	
Kontravalenz	XOR	$A \oplus B$	$A \neq B$	$(A \uparrow (A \uparrow B)) \uparrow (B \uparrow (A \uparrow B))$	

Tabelle 1: Die wichtigsten Junktoren der Schaltalgebra.

### 1.1.2. Elementare Identitäten

Um mit der Schaltalgebra effizient rechnen zu können, sind die folgenden Äquivalenzen (Tautologien) von großem Nutzen (wir schreiben kurz für  $\equiv$  auch  $=$ ):

- **Kommutativität:**  $A \cdot B = B \cdot A$  und  $A + B = B + A$ .
- **Assoziativität:**  $(A \cdot B) \cdot C = A \cdot (B \cdot C)$  und  $(A + B) + C = A + (B + C)$ .
- **Distributivität:**  $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$  und  $A + (B \cdot C) = (A + B) \cdot (A + C)$ .
- **Idempotenz:**  $A \cdot A = A$  und  $A + A = A$ .
- **Absorptionsgesetze:**  $A \cdot (A + B) = A + A \cdot B = A$ .
- **De Morgansche Regeln:**  $\overline{A \cdot B} = \overline{A} + \overline{B}$  und  $\overline{A + B} = \overline{A} \cdot \overline{B}$ .
- **XOR:**  $A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$  und  $A \cdot (B \oplus C) = (A \cdot B) \oplus (A \cdot C)$ .

Desweiteren gelten die Äquivalenzen aus Tabelle 1 bezüglich der NAND-Funktion (Schefferstrich  $\uparrow$ ). Die hier aufgelisteten Äquivalenzen können oft bei der Planung logischer Schaltungen verwendet werden, um deren Komplexität zu reduzieren und so den Bauteilebedarf, den Stromverbrauch sowie die Reaktionszeiten zu optimieren.

### 1.1.3. Elektrotechnische Realisierung

Realisiert man obige logische Funktionen mit elektronischen Schaltkreisen ist man gezwungen den unterschiedlichen Spannungspegeln, die meist durch ein Spannungsintervall undefinierter Zustände voneinander getrennt sind, entsprechende logische Zustände zuzuordnen. Liegt an einem Messpunkt ein hohes Potential an, wird dieser Zustand mit H, bei einem niedrigen Potential mit L bezeichnet. Welches Potential als H und welches als L zu interpretieren ist, wird durch die Spezifikation der verwendeten Logik-Familie festgelegt. Es gibt folglich zwei mögliche Zuordnungen, welche die Verbindung zwischen elektrotechnischer Realisierung und logischer Bedeutung herstellen:

- **positive Logik:** Hier gelten die Zuordnungen  $H \leftrightarrow 1$  und  $L \leftrightarrow 0$ .
- **negative Logik:** Hier gelten die Zuordnungen  $H \leftrightarrow 0$  und  $L \leftrightarrow 1$ .

Je nach Interpretation der Ein- und Ausgangspegel kann eine physikalische Implementierung *verschiedene* logische Funktionen realisieren. Beispielsweise realisiert eine Schaltung, die in positiver Logik als OR-Gatter fungiert, in negativer Logik ein AND-Gatter und umgekehrt. Diese funktionale Dualität spiegelt sich in der Logik durch die De Morganschen Regeln wieder.

Logische Schaltungen werden in der Elektrotechnik durch *logische Gatter* realisiert, die meist als IC geliefert werden und Mitglied einer Logikfamilie sind. Die Bauteile einer Logikfamilie sind hinsichtlich Versorgungsspannung, Ein- und Ausgangspegeln aufeinander abgestimmt und lassen sich daher in gewissen Grenzen (*Fan-out*) ohne Verstärker oder weitere Beschaltung zu Logiknetzwerken verschalten. Eine weitverbreitete Logikfamilie ist die sog. *Transistor-Transistor-Logik* (kurz **TTL**), die auch in den folgenden Versuchen eingesetzt wird. Die TTL-NAND-Gatter werden aus einem Ein- und einem Ausgangstransistor mit entsprechender Beschaltung aufgebaut (daher der Name) und realisieren für positive Logik eine NAND-Funktion. Die Funktionstabelle eines solchen Gatters ist in Tabelle 2 (links) abgebildet.

<sup>1</sup>Symbole der Schaltalgebra

<sup>2</sup>Symbole der Logik

### 1.2. Zahlensysteme

Um das mathematisch abstrakte Objekt einer ‘Zahl’ alltagstauglich darstellen zu können, bedient man sich sog. *Zahlensysteme*. Während die zumeist primitiven *Additionssysteme* (vgl. Strichlisten  $\Leftrightarrow$  Unärsystem) nur für Zahlen kleiner Beträge handhabbar sind, erlauben *Stellenwertsysteme* die Abdeckung weiter Zahlenbereiche. Der Preis für diese neue Eigenschaft ist das kompliziertere Additionsverfahren. Stellenwertsysteme zeichnen sich durch einen Satz von  $b \geq 2$  Symbolen aus und werden dann als *b-adisches Zahlensystem* mit der Grundzahl  $b$  bezeichnet. Verbreitete Basen sind

$$\mathcal{B}_2 = \{0, 1\}, |\mathcal{B}_2| = 2 = \mathbf{b} \quad \text{Dualsystem} \quad (1)$$

$$\mathcal{B}_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}, |\mathcal{B}_8| = 8 = \mathbf{b} \quad \text{Oktalsystem} \quad (2)$$

$$\mathcal{B}_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, |\mathcal{B}_{10}| = 10 = \mathbf{b} \quad \text{Dezimalsystem} \quad (3)$$

$$\mathcal{B}_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}\}, |\mathcal{B}_{16}| = 16 = \mathbf{b} \quad \text{Hexadezimalsystem} \quad (4)$$

Dabei sind die **fett** gekennzeichneten Zahlen und Buchstaben als **Symbole**  $a_i \in \mathcal{B}_b$  zu interpretieren, die eineindeutig mit *Zahlen*  $a_i^* \in \mathbb{N}$  identifiziert werden. Eine Ziffernfolge  $(a_i)_{i=0}^n \subset \mathcal{B}_b$  wird dann wie folgt mit einer Zahl  $N \in \mathbb{N}$  identifiziert:

$$N = [(a_i)_{i=0}^n]_b = \sum_{i=0}^n a_i^* \cdot b^i \quad (5)$$

Da  $2 = 2^1$ ,  $8 = 2^3$  und  $16 = 2^4$  werden Dual-, Oktal und Hexadezimalsysteme bei informationsverarbeitenden digitalen Systemen verwendet, die auf Grund der Implementierung nur mit der zweiwertigen booleschen Algebra umgehen können. Folglich lassen sich auf einer solchen Maschine die Ziffern von Dualzahlen mit 1 Bit, von Oktalzahlzahlen mit 3 Bit und von Hexadezimalzahlen mit 4 Bit ohne Vergeudung von Speicher darstellen. Beispiele für die Darstellung dezimal gegebener Zahlen bezüglich der Basis  $b = 2$  sind in Tabelle 9 gegeben.

### 1.3. Beispielrechnungen

Die in Abb. 3 (links) dargestellte Schaltung aus NAND-Gattern lässt sich mit den Regeln der Schaltalgebra wie folgt ausdrücken:

$$A_3(E_1, E_2) = \overline{\overline{E_1} \cdot \overline{E_1} \cdot \overline{E_1} \cdot \overline{E_1}} = \overline{\overline{E_1} \cdot \overline{E_1}} = \overline{\overline{E_1}} + \overline{\overline{E_1}} = E_1 + E_1 \quad (6)$$

Damit realisiert eine Schaltung dieser Struktur aus NAND-Gattern für positive Logik eine OR-Funktion. Interpretiert man die physikalische Schaltung in negativer Logik, also indem  $H \leftrightarrow 0$  und  $L \leftrightarrow 1$  identifiziert werden, entspricht dies der logischen Inversion aller Signale an Ein- und Ausgängen:

$$A'_3(E'_1, E'_2) = \overline{\overline{\overline{E_1}} \cdot \overline{\overline{E_2}}} = \overline{\overline{E_1}} + \overline{\overline{E_2}} = \overline{\overline{E_1} \cdot \overline{E_1}} = E_1 \cdot E_1 \quad (7)$$

Damit wird eine AND-Funktion durch *die selbe physikalische Schaltung* repräsentiert, indem deren logische Pegel in negativer Logik interpretiert werden.

Die in Abb. 3 (rechts) dargestellte Implementierung der XOR-Funktion lässt sich mit den Regeln der Schaltalgebra wie folgt auflösen:

$$\begin{aligned} A_4(E_1, E_2) &= \overline{\overline{\overline{E_1} \cdot \overline{E_1} \cdot \overline{E_2} \cdot \overline{E_2} \cdot \overline{E_1} \cdot \overline{E_2}}} \\ &= \overline{\overline{E_1} \cdot (\overline{E_1} + \overline{E_2}) \cdot \overline{E_2} \cdot (\overline{E_1} + \overline{E_2})} \\ &= \overline{\overline{E_1} \cdot \overline{E_1} + \overline{E_1} \cdot \overline{E_2} \cdot \overline{E_2} \cdot \overline{E_2} + \overline{E_2} \cdot \overline{E_1}} \\ &= \overline{\overline{E_1} \cdot \overline{E_2} \cdot \overline{E_2} \cdot \overline{E_1}} \\ &= (\overline{E_1} \cdot \overline{E_2}) + (\overline{E_2} \cdot \overline{E_1}) = \text{XOR}(E_1, E_2) \end{aligned}$$

Die in Abb. 4 dargestellte logische Struktur realisiert einen 2-zu-1-Multiplexer. Die Schaltung beinhaltet allerdings ein NOT-, zwei AND- sowie ein OR-Gatter. Da wir nur über NAND-Gatter verfügen, dieses aber ein SHEFFER-Operator ist, lassen sich alle in obiger Schaltung verwendeten Funktionen durch NAND-Gatter ausdrücken. Die Reduktion um zwei doppelte Negationen ergibt dann die logische Schaltung in Abb. 1. Diese wird beim entsprechenden Versuch auch benutzt, um den Multiplexer zu implementieren.

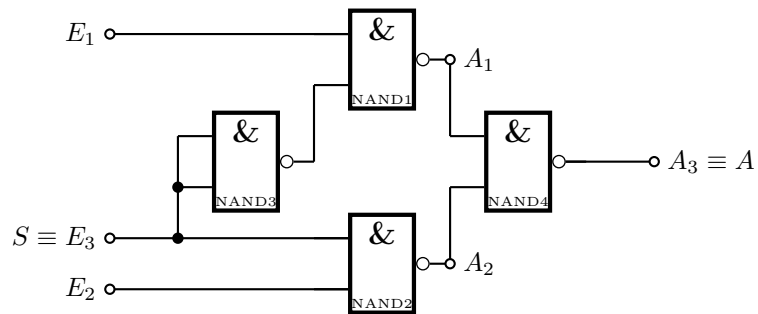


Abbildung 1: Auf NAND-Gatter reduzierter Schaltplan des 2-zu-1-Multiplexers.

## 2. Messprinzip, Schaltpläne und Versuchsablauf

### 2.1. Eigenschaften von TTL-Logikbausteinen

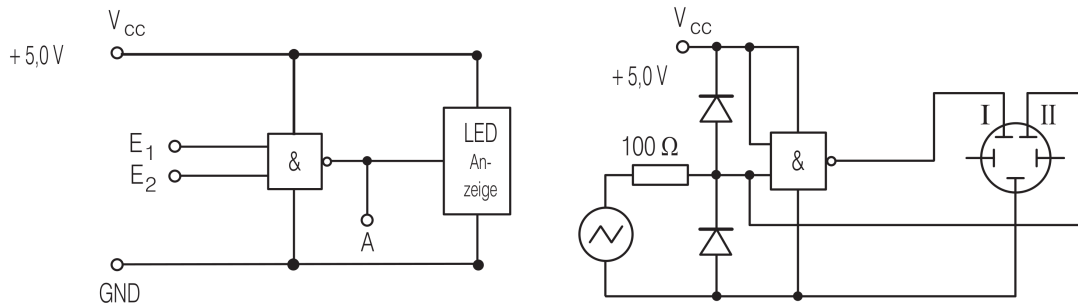


Abbildung 2: Schaltplan zur Ermittlung der Funktionstabelle (links) und zur Ausmessung der Schaltschwellen (rechts).

**Messprinzip und Versuchsaufbau** Die Schaltung wird nach Abbildung 2 (links) aufgebaut. Als NAND-Gatter kommt der TTL-Baustein DM7400 für positive Logik zum Einsatz. An den Ausgang  $A$  wird das Tischmultimeter (Gleichspannungsmessung) angeschlossen. Vor Eingang  $E_1$  wird ein weiteres Tischmultimeter (Gleichstrommessung) in Reihe geschaltet. Zur Anzeige der Potentiallevel werden zweifarbige LEDs mit entsprechender Beschaltung benutzt. Für den zweiten Teil des Versuches wird die Schaltung gemäß Abb. 2 (rechts) aufgebaut. Der Funktionsgenerator wird dabei über den Sync-Ausgang mit dem Trigger-Eingang des Oszilloskops verbunden.

**Versuchsablauf** Im ersten Schritt wird die Funktionstabelle aller vier integrierter NAND-Gatter mit Hilfe der LED-Anzeige aufgenommen. Hierfür werden die Eingänge mit  $GND$  bzw.  $V_{CC}$  verbunden. Die Spannungen am Ausgang werden für alle vier Zustände gemessen und notiert. Analog werden die Eingangsströme exemplarisch für ein Gatter und einen Eingang in Abhängigkeit vom Gatterzustand gemessen.

Im zweiten Teil des Versuches wird ein Eingang auf  $V_{CC}$  gelegt (siehe Abb. 2 (rechts)). Der Funktionsgenerator wird auf Dreiecksignal mit  $\nu \in \{50, 10\} \text{Hz}$ ,  $U_{e,pp} = 5.0 \text{V}$  sowie  $U_{e,off} = 2.5 \text{V}$  gestellt. Die Signalverläufe von CH1 und CH2 werden über LabVIEW abgespeichert.

**Geräte** DM7400 Quad-TTL-NAND-Gatter, Halbleiterdiode (2x),  $100\Omega$ -Widerstand, Keithley 2100 (Tischmultimeter, 2x), Keithley 3390 (Funktionsgenerator), Tektronix (Digitaloszilloskop, 2 Kanäle), LED-Modul, Steckbrett mit Spannungsquelle. ◀

### 2.2. NAND als elementares Gatter

**Messprinzip und Versuchsaufbau** Im ersten Teil des Versuches wird eine OR-Funktion nach Abb. 3 (links) aufgebaut. Im zweiten Teil wird die Implementierung der XOR-Funktion nach Abb. 3 (rechts) aufgebaut und anschließend zu einem Halbaddierer ergänzt (siehe Auswertung). Die Messpunkte  $A_i$  werden jeweils mit einer LED des LED-Moduls verbunden.

**Versuchsablauf** Sowohl für die OR- als auch die XOR-Funktion werden die logischen Level an den Messpunkten  $A_i$  für alle vier Zustände mit Hilfe der LED-Anzeige gemessen. Der Halbaddierer wird analog untersucht, indem die Level an den Ausgängen  $S$  (Summe) und  $C$  (Carry-Bit) in Abhängigkeit vom Eingangszustand gemessen werden.

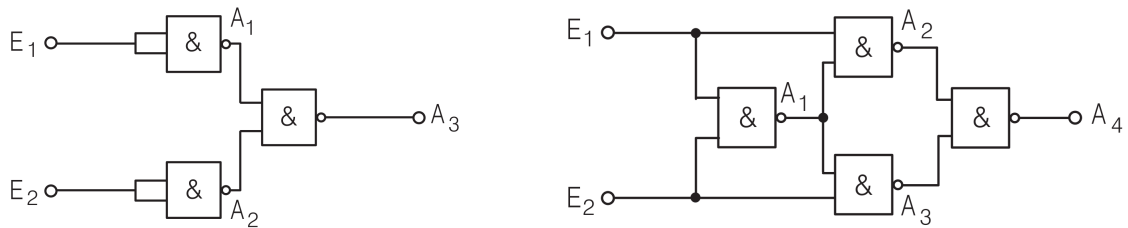


Abbildung 3: Schaltplan zur Realisierung einer OR-Funktion (links) sowie einer XOR-Funktion (rechts).

**Geräte** DM7400 Quad-TTL-NAND-Gatter (2x), LED-Modul, Steckbrett mit Spannungsquelle. ◀

### 2.3. Multiplexer

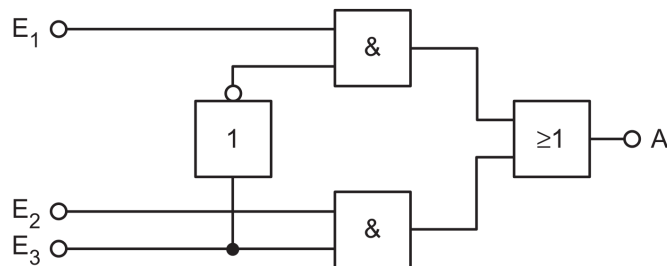


Abbildung 4: Schaltplan eines 2-nach-1-Multiplexers.

**Messprinzip und Versuchsaufbau** Der 2-nach-1-Multiplexer wird gemäß Abb. 4 aus NAND-Gattern aufgebaut (für die Realisierung mit NAND-Gattern siehe auch Abb. 1).  $A$  wird mit einer LED des LED-Moduls verbunden.

**Versuchsablauf** Die Funktionstabelle der Schaltung wird aufgenommen, indem alle  $2^3 = 8$  Zustände untersucht werden. Mit Hilfe der Tabelle wird der Steuereingang  $S$  identifiziert.

**Geräte** DM7400 Quad-TTL-NAND-Gatter, LED-Modul, Steckbrett mit Spannungsquelle. ◀

### 2.4. Dekodierschaltungen

**Messprinzip und Versuchsaufbau** Der 2-Bit-Dekoder wird nach Abb. 5 aus NAND-Gattern aufgebaut. Die Ausgänge 0, 1, 2 und 3 werden mit je einer LED verbunden.

**Versuchsablauf** Die Funktionstabelle der Schaltung wird aufgenommen, indem alle  $2^2 = 4$  Zustände untersucht werden.

**Geräte** DM7400 Quad-TTL-NAND-Gatter (2x), LED-Modul, Steckbrett mit Spannungsquelle. ◀

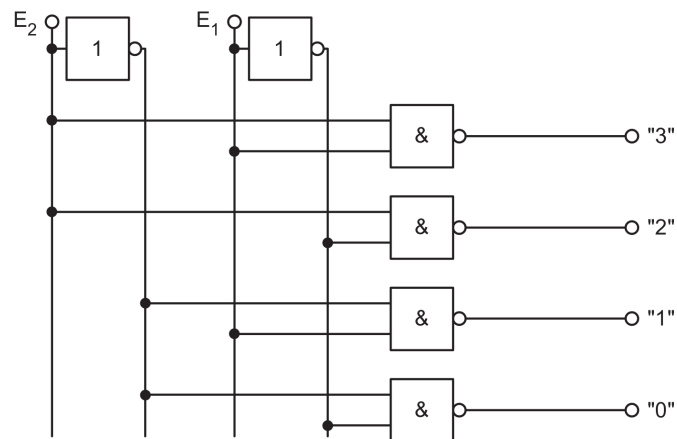


Abbildung 5: Schaltplan eines 2-Bit-Dekoders für Dualzahlen.

### 3. Berechnungen & Formeln

Die wichtigsten logischen Identitäten wurden in den Grundlagen behandelt. Auf Grund der qualitativen Natur der Versuche sind keine weiteren formalen Zusammenhänge nötig.

$E_1$	$E_2$	$A$	$I_{E_1}$	$U_A$
H	H	L	$0.1\mu\text{A}$	$143.7\text{mV}$
L	L	H	$-63.1\mu\text{A}$	$3.93\text{V}$
L	H	H	$-220.2\mu\text{A}$	$3.93\text{V}$
H	L	H	$0.1\mu\text{A}$	$3.93\text{V}$

$E_1$	$E_2$	$A$
1	1	0
0	0	1
0	1	1
1	0	1

Tabelle 2: Funktionstabelle und Kenngrößen des TTL-NAND-Gatters (links) sowie die zugehörige Wahrheitstabelle in positiver Logik (rechts).

## 4. Auswertung

### 4.1. Eigenschaften von TTL-Logikbausteinen

#### 4.1.1. Funktionsweise eines TTL-Gatters DM7400

Um die Funktionstüchtigkeit der 4 NAND-Gatter des DM7400-ICs zu überprüfen, wurden die Funktionstabellen jedes Gatters ermittelt. Da alle 4 Gatter unbeschädigt waren, ergab sich jeweils Tabelle 2 (links) bzw. in positiver Logik die Wahrheitstafel von Tabelle 2 (rechts). Der Tabelle zufolge wird in positiver Logik eine NAND-Funktion realisiert. Die Potentiallevel der unterschiedlichen logischen Zustände wurde zu  $U_{LOW} \approx 0.1\text{V}$  sowie  $U_{HIGH} \approx 3.9\text{V}$  bestimmt. Dem Datenblatt zufolge ist  $U_{LOW} \leq 0.4\text{V}$  mit  $U_{LOW}^{typ} = 0.2\text{V}$  und  $U_{HIGH} \geq 2.4\text{V}$  mit  $U_{HIGH}^{typ} = 3.4\text{V}$ . Unsere Messwerte liegen daher im Rahmen der Spezifikationen.

Die Gattereingangsströme wurden exemplarisch an Eingang  $E_1$  eines der Gatter gemessen. Die Werte sind in Tab. 2 (links) gelistet. Wenn der betrachtete Eingang auf H liegt, fließen nur minimale Ströme von  $I_{E_1}^{HIGH} \approx 0.1\mu\text{A}$  in den Eingang hinein. Das Datenblatt garantiert  $I_{E_1}^{HIGH} \leq 40\mu\text{A}$ . Liegt  $E_1$  hingegen auf L und  $E_2$  auf H fließt ein Strom von  $I_{E_1}^{LOW,H} \approx -220\mu\text{A}$  aus dem Eingang heraus während für  $E_2$  ebenfalls auf L nur noch ein Teil des Stromes, nämlich  $I_{E_1}^{LOW,L} \approx -60\mu\text{A}$  aus  $E_1$  herausfließt. Das Datenblatt garantiert für beide Fälle  $|I_{E_1}^{LOW}| \leq 1.6\text{mA}$ . Damit liegen unsere Messwerte im Bereich der spezifizierten Werte.

#### 4.1.2. Schaltschwellen des TTL-Gatters DM7400

Wird der Eingang  $E_2$  auf H gesetzt während  $E_1$  mit einem Dreiecksignal mit  $0.0\text{V} \leq U_e(t) \leq 5.0\text{V}$  belegt wird, ergeben sich Zeitverläufe der Ein- und Ausgangsspannungen wie in den Abbildungen 6 und 7 dargestellt. Da  $E_2$  fest auf H liegt, kann der Ausgang nur auf H liegen während  $E_1$  auf L liegt. Demzufolge springt der Ausgang im Bereich kleinster Eingangsspannungen auf H.

Trägt man die Ausgangsspannung  $U_a$  gegen die Eingangsspannung  $U_e$  auf, ergibt sich die in Abb. 8 dargestellte Grafik. Dem Datenblatt zufolge wird ein Eingangspegel von  $U_e < 0.8\text{V}$  garantiert als L und ein Pegel von  $U_e > 2.0\text{V}$  mit Sicherheit als H erkannt. Die vertikalen Linien bezeichnen diese Grenzen. Für den Ausgang garantieren die Spezifikationen  $U_a > 2.4\text{V}$  für H und  $U_a < 0.4\text{V}$  für L<sup>3</sup>. Diese Grenzen werden durch die horizontalen Linien dargestellt. Demnach dürfen dem Datenblatt zufolge nur in den grau unterlegten Bereichen Zustände gemessen werden, wobei die dunkelgrauen Zonen definierte Ein- und Ausgangszustände darstellen während die hellgraue Zone den Bereich eines undefinierten Eingangspegels überdeckt.

Unsere Messpunkte liegen bis auf wenige Ausnahmen in den legitimen Bereichen. Wie man an den Plots in den Abbildungen 6 und 7 erkennt, sind sowohl Ein- als auch Ausgangsspannung in den Bereichen undefinierter Zustände starken Fluktuationen (z.B. durch Einstreuungen) unterworfen. Die Fluktuationen der Eingangsspannung sind der Grund für die links aus der hellgrauen Zone undefinierter Zustände liegenden Messpunkte.

<sup>3</sup>Dies gilt nur für schwach belastete Ausgänge. Dies war in unserem Fall gewährleistet.

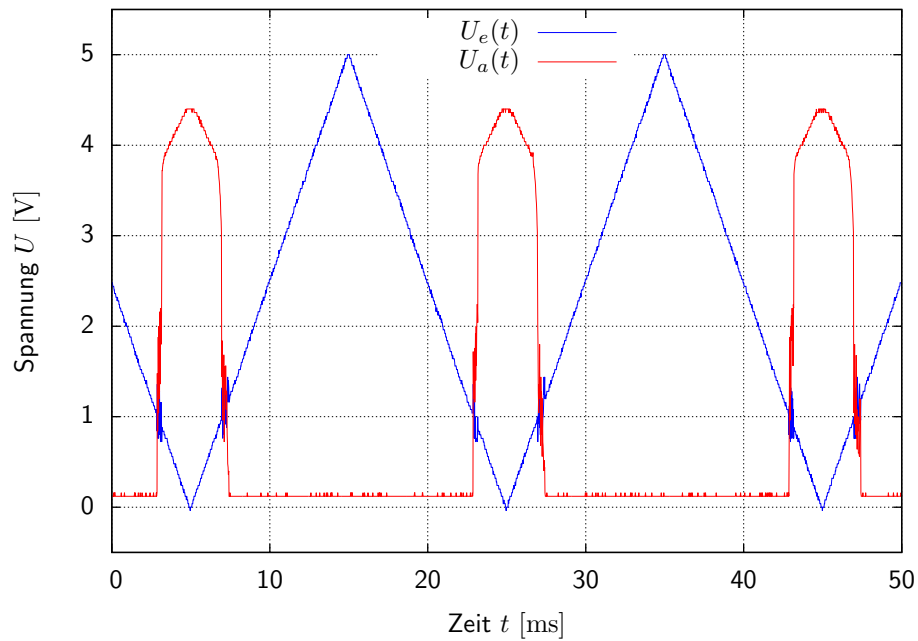


Abbildung 6: Eingangsspannung  $U_e$  und Ausgangsspannung  $U_a$  des TTL-NAND-Gatters gegen die Zeit aufgetragen ( $\nu = 50\text{Hz}$ ,  $U_{e,pp} = 5.0\text{V}$ ,  $U_{e,off} = 2.5\text{V}$ ).

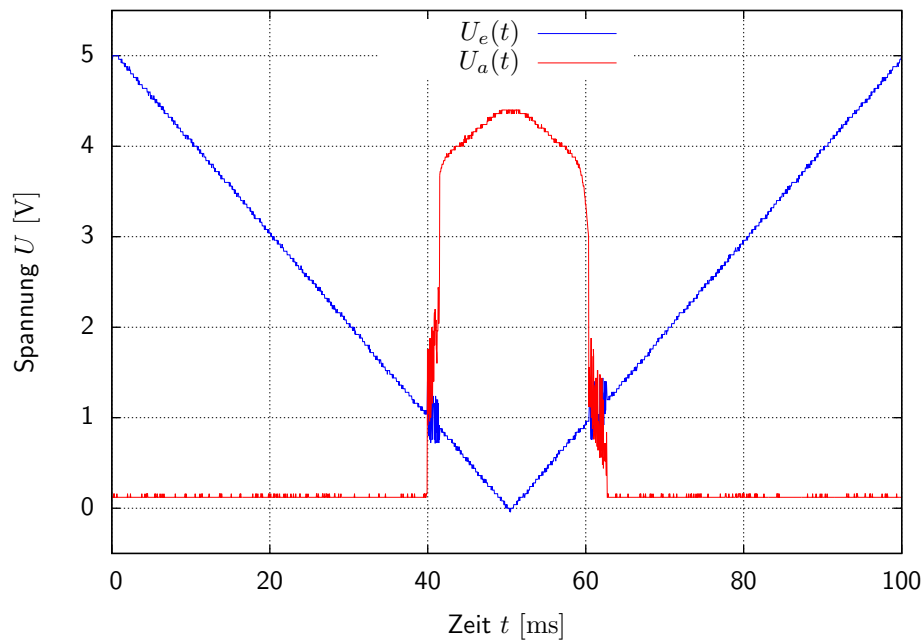


Abbildung 7: Eingangsspannung  $U_e$  und Ausgangsspannung  $U_a$  des TTL-NAND-Gatters gegen die Zeit aufgetragen ( $\nu = 10\text{Hz}$ ,  $U_{e,pp} = 5.0\text{V}$ ,  $U_{e,off} = 2.5\text{V}$ ).

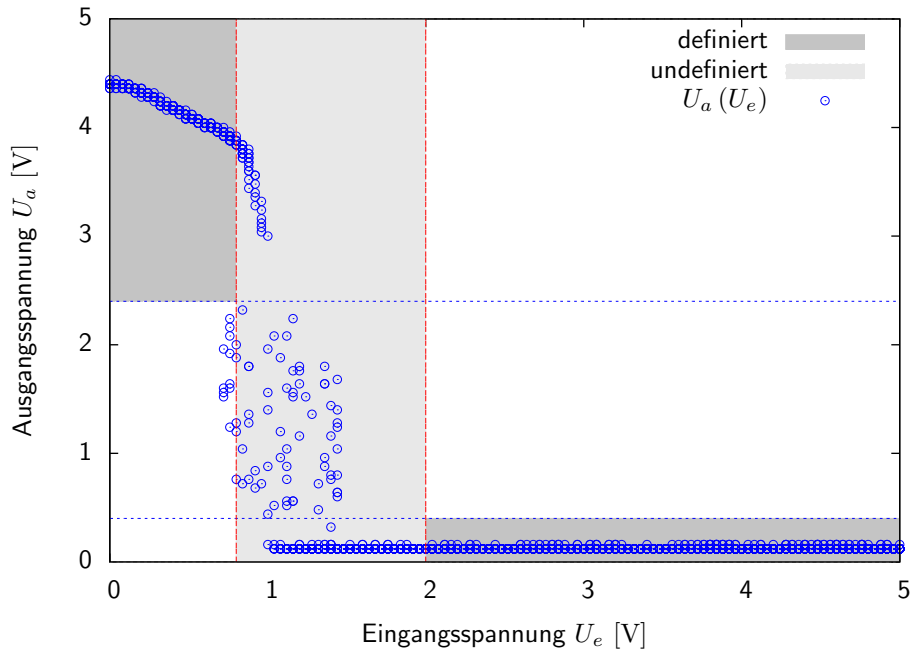


Abbildung 8: Ausgangsspannung  $U_a$  gegen die Eingangsspannung  $U_e$  aufgetragen ( $\nu = 50\text{Hz}$ ,  $U_{e,pp} = 5.0\text{V}$ ,  $U_{e,off} = 2.5\text{V}$ ).

$E_1$	$E_2$	$A_1$	$A_2$	$A_3$	$E_1$	$E_2$	$A_1$	$A_2$	$A_3$	$E_1$	$E_2$	$A_1$	$A_2$	$A_3$
H	H	L	L	<b>H</b>	1	1	0	0	<b>1</b>	0	0	1	1	<b>0</b>
L	L	H	H	<b>L</b>	0	0	1	1	<b>0</b>	1	1	0	0	<b>1</b>
L	H	H	L	<b>H</b>	0	1	1	0	<b>1</b>	1	0	0	1	<b>0</b>
H	L	L	H	<b>H</b>	1	0	0	1	<b>1</b>	0	1	1	0	<b>0</b>

Tabelle 3: Funktionstabelle der Implementierung des OR-Gatters (links) sowie die logische Interpretation in positiver (mitte) und negativer (rechts) Logik.

## 4.2. NAND als elementares Gatter

### 4.2.1. OR-Funktion aus NAND-Gattern

Die Funktionstabelle in Tab. 3 (links) zeigt die gemessenen Potentiallevel für alle vier Zustände. Hierbei steht H für “High-Level” und L für “Low-Level”. Interpretiert man diese Schaltung in positiver Logik ergibt sich die Wahrheitstafel in Tab. 3 (mitte) mit den Entsprechungen  $H \leftrightarrow 1$  und  $L \leftrightarrow 0$ . Diese Wahrheitstafel definiert eine logische OR-Funktion. An den Messpunkten  $A_1$  und  $A_2$  findet offensichtlich eine Negation des jeweiligen Eingangs statt.

Interpretiert man die gemessene Funktionstabelle in negativer Logik, also mit  $H \leftrightarrow 0$  und  $L \leftrightarrow 1$ , so folgt die Wahrheitstafel in Tab. 3 (rechts). Diese beschreibt eine AND-Funktion. Dieses Ergebnis ist Folge der Dualität zwischen OR und AND in positiver bzw. negativer Logik.

### 4.2.2. XOR-Funktion aus NAND-Gattern

Die Funktionstabelle in Tab. 4 (links) zeigt die gemessenen Ausgangszustände für alle vier Eingangszustände des Gatters. Interpretiert in positiver Logik (siehe Tab. 4 (rechts)) ergibt sich die Wahrheitstafel der XOR-Funktion – in der Logik auch als Antivalenz bezeichnet.

$E_1$	$E_2$	$A_1$	$A_2$	$A_3$	$A_4$
H	H	L	H	H	<b>L</b>
L	L	H	H	H	<b>L</b>
L	H	H	H	L	<b>H</b>
H	L	H	L	H	<b>H</b>

$E_1$	$E_2$	$A_1$	$A_2$	$A_3$	$A_4$
1	1	0	1	1	<b>0</b>
0	0	1	1	1	<b>0</b>
0	1	1	1	0	<b>1</b>
1	0	1	0	1	<b>1</b>

Tabelle 4: Funktionstabelle der Implementierung der XOR-Funktion (links) und die zugehörige Wahrheitstafel in positiver Logik (rechts).

$E_1$	$E_2$	$S$	$C$
H	H	L	H
L	L	L	L
L	H	H	L
H	L	H	L

$E_1$	$E_2$	$S$	$C$
1	1	0	1
0	0	0	0
0	1	1	0
1	0	1	0

Tabelle 5: Funktionstabelle der Implementierung des Halbaddierers (links) und die zugehörige Wahrheitstafel in positiver Logik (rechts).

Die hier realisierte XOR-Funktion errechnet die Summe  $S$  bei der Addition einer 1-Bit Dualzahl (i.e. ohne Berücksichtigung eines Übertrages einer vorangegangenen Addition). Um einen vollständigen Halbaddierer zu erhalten muss ein Überlauf des 1-Bit Registers ( $1 \oplus 1 = 0$ ) registriert und mit Hilfe eines sog. *Carry-Bits*  $C$  an den u.U. folgenden Volladdierer weitergereicht werden. Hierfür wird dem schon aufgebauten XOR-Gatter, wie in Abb. 9 dargestellt, ein AND-Gatter parallel geschaltet.

In Tabelle 5 ist die Funktionstabelle (links) und die zugehörige Wahrheitstafel in positiver Logik (rechts) dargestellt. Wie erwartet signalisiert das Carry-Bit ("Übertrag") die Summe  $1 \oplus 1 = 0$  während die Ergebnisse der Summen  $1 \oplus 0 = 1$ ,  $0 \oplus 1 = 1$  und  $0 \oplus 0 = 0$  alleine am Bit  $S$  ("Summe") abgelesen werden können.

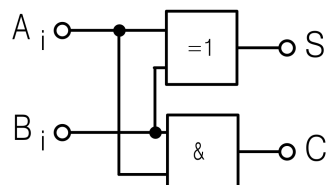


Abbildung 9: Schaltplan eines Halbaddierers für Dualzahlen.

### 4.3. Multiplexer

Die Aufgabe eines Multiplexers besteht in der über einen binär codierten Schalter  $S$  kontrollierten Weiterleitung eines anliegenden Signals. Alle übrigen anliegenden Signale (Eingänge) werden dabei blockiert. Der hier verwendete 2-nach-1-Multiplexer wählt aus zwei anliegenden Signalen je nach Schalterstellung eines aus und leitet dieses an den Ausgang  $A$  weiter.

In Tabelle 6 sind die Funktionstabelle und die Wahrheitstafel in positiver Logik der untersuchten Schaltung dargestellt. In der (willkürlich) gewählten Reihenfolge der  $2^3 = 8$  gewählten Eingangszustände ist zunächst keine Struktur erkennbar. Sortiert man die Zeilen aber neu erhält man die Tabellen in Tab. 7. Hier erkennt man sofort, dass Eingang  $E_3 \equiv S$  als Steuereingang fungiert. Je nach logischem Zustand wird entweder das Signal von  $E_1$  oder das von  $E_2$  an den Ausgang

$E_1$	$E_2$	$E_3 \equiv S$	$A_1$	$A_2$	$A_3 \equiv A$
H	H	H	H	L	<b>H</b>
H	H	L	L	H	<b>H</b>
H	L	H	H	H	<b>L</b>
H	L	L	L	H	<b>H</b>
L	H	H	H	L	<b>H</b>
L	H	L	H	H	<b>L</b>
L	L	H	H	H	<b>L</b>
L	L	L	H	H	<b>L</b>

$E_1$	$E_2$	$E_3 \equiv S$	$A_1$	$A_2$	$A_3 \equiv A$
1	1	1	1	0	<b>1</b>
1	1	0	0	1	<b>1</b>
1	0	1	1	1	<b>0</b>
1	0	0	0	1	<b>1</b>
0	1	1	1	0	<b>1</b>
0	1	0	1	1	<b>0</b>
0	0	1	1	1	<b>0</b>
0	0	0	1	1	<b>0</b>

Tabelle 6: Funktionstabelle der Implementierung des Multiplexers (links) und die zugehörige Wahrheitstafel in positiver Logik (rechts).

$E_1$	$E_2$	$E_3 \equiv S$	$A_1$	$A_2$	$A_3 \equiv A$
H	<b>H</b>	H	H	L	<b>H</b>
L	<b>H</b>	H	H	L	<b>H</b>
H	<b>L</b>	H	H	H	<b>L</b>
L	<b>L</b>	H	H	H	<b>L</b>
<b>H</b>	H	L	L	H	<b>H</b>
<b>H</b>	L	L	L	H	<b>H</b>
<b>L</b>	H	L	H	H	<b>L</b>
<b>L</b>	L	L	H	H	<b>L</b>

$E_1$	$E_2$	$E_3 \equiv S$	$A_1$	$A_2$	$A_3 \equiv A$
1	<b>1</b>	1	1	0	<b>1</b>
0	<b>1</b>	1	1	0	<b>1</b>
1	<b>0</b>	1	1	1	<b>0</b>
0	<b>0</b>	1	1	1	<b>0</b>
<b>1</b>	1	0	0	1	<b>1</b>
<b>1</b>	0	0	0	1	<b>1</b>
<b>0</b>	1	0	1	1	<b>0</b>
<b>0</b>	0	0	1	1	<b>0</b>

Tabelle 7: Funktionstabelle der Implementierung des Multiplexers (links) und die zugehörige Wahrheitstafel in positiver Logik (rechts) [umsortiert].

$A \equiv A_3$  übertragen.

Die logischen Zustände an den Messpunkten  $A_1$  und  $A_2$  wurden zur Kontrolle aufgenommen und stellen die Zustände nach den beiden Eingangsgattern dar.

#### 4.4. Dekodierschaltungen

Der 2-Bit-Dekoder ergab die in Tab. 8 (links) dargestellte Funktionstabelle. Um die gewünschte Dekodierung der am Eingang anliegenden 2-Bit-Dualzahl zu erreichen muss das Low-Level am Ausgang als aktiver Zustand interpretiert werden ( $L \leftrightarrow \otimes$ ) während die High-Level inaktive Zustände darstellen ( $H \leftrightarrow \circ$ ). Am Eingang müssen die Level in positiver Logik interpretiert werden um eine korrekte Dekodierung zu erhalten – also  $H \leftrightarrow 1$  und  $L \leftrightarrow 0$ . Desweiteren ist zu beachten, dass Eingang  $E_1$  dem niederwertigen Bit und  $E_2$  dem höherwertigen Bit entspricht. Demnach müssen zur korrekten Darstellung der Dualzahl die Spalten  $E_1$  und  $E_2$  vertauscht werden.

Der Treiber einer 7-Segment-Anzeige, welcher heute i.d.R. als IC vorliegt, besteht aus einem

$E_1$	$E_2$	0	1	2	3
L	L	<b>L</b>	H	H	H
H	L	H	<b>L</b>	H	H
L	H	H	H	<b>L</b>	H
H	H	H	H	H	<b>L</b>

$E_2$	$E_1$	0	1	2	3
0	0	$\otimes$	$\circ$	$\circ$	$\circ$
0	1	$\circ$	$\otimes$	$\circ$	$\circ$
1	0	$\circ$	$\circ$	$\otimes$	$\circ$
1	1	$\circ$	$\circ$	$\circ$	$\otimes$

Tabelle 8: Funktionstabelle der Implementierung des 2-Bit Dekoders (links) und eingangsseitige Interpretation in positiver Logik (rechts).

Dezimal	Binär	Segmente						
		a	b	c	d	e	f	g
0	0000	1	1	1	1	1	1	0
1	0001	0	1	1	0	0	0	0
2	0010	1	1	0	1	1	0	1
3	0011	1	1	1	1	0	0	1
4	0100	0	1	1	0	0	1	1
5	0101	1	0	1	1	0	1	1
6	0110	1	0	1	1	1	1	1
7	0111	1	1	1	0	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	1	0	1	1

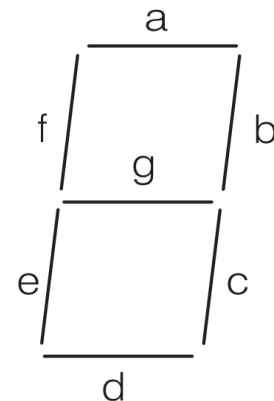


Tabelle 9: Funktionstabelle (links) und schematische Zeichnung (rechts) der 7-Segment Anzeige.

1-aus-4-Decoder mit angeschlossenem Festwertspeicher. Der Decoder aktiviert je nach anliegender 4-Bit-Dualzahl einen der 10 Zweige des Festwertspeichers (für Ziffern). Der Festwertspeicher gibt entsprechend des aktivierten Einganges ein 7-Bit-Signal aus, welches die Segmente der Anzeige ansteuert. Dies geschieht nach dem in Tabelle 9 dargestellten Musters, das zugleich die im Festwertspeicher zu codierende Matrix darstellt.

Die 4-Bit-Codierung ist nötig, da  $2^3 = 8 < 10$  ist und eine 3-Bit-Dualzahl damit zu wenig Information für das Dezimale 10-Ziffern-System tragen kann. Wegen  $2^4 = 16 = 10 + 6$  werden bei der Codierung von Dezimalzahlen 6 Dualzahlen nicht zum Informationstransport genutzt. Heutige Treiber von 7-Segmentanzeigen codieren daher die Ziffern von Hexadezimalzahlen, da diese alle Kombinationsmöglichkeiten der 4 Bit nutzen. Die Anzeigen können daher neben den Ziffern 0 – 9 auch noch die Buchstaben A – F darstellen.

## 5. Fehlerrechnung

Da die durchgeführten Versuche alle qualitativer bzw. digitaler Natur waren, erübrigt sich die Betrachtung von Messfehlern.

## 6. Zusammenfassung

**Eigenschaften von TTL-Logikbausteinen** Der DM7400-IC stellt in positiver Logik 4 NAND-Gatter zur Verfügung. Dies konnte durch Messen der Pegel am Ausgang für die vier Eingangszustände verifiziert werden. Dabei wurde das High-Level zu  $U_{HIGH} \approx 3.9V$  und das Low-Level zu  $U_{LOW} \approx 0.1V$  bestimmt. Die Eingangsströme hängen vom Eingangszustand ab und sind für einen Eingang auf H sehr niedrig ( $I_{E_1}^{HIGH} \approx 0.1\mu A$ ) während sie für einen Eingang auf L deutlich größer ( $I_{E_1}^{LOW,H} \approx -220\mu A$ ) ausfallen. Dabei hängt der aus dem Eingang herausfließende Strom (Eingang auf L) vom Pegel am anderen Eingang ab während der in den Eingang hineinfließende Strom (Eingang auf H) im Rahmen unserer Messung unabhängig vom Zustand des anderen Eingangs war.

Die spezifizierten Bereiche für definierte logische Zustände konnten verifiziert werden. Dabei wurde für Spannungen  $U_e < 0.8V$  der Eingang korrekt als L klassifiziert woraufhin das Signal am Ausgang mit  $U_a > 2.4V$  als H gesetzt wurde. Umgekehrt wurden Spannungen  $U_e > 2.0V$  stabil als H erkannt und daraufhin der Ausgang mit  $U_a < 0.4V$  auf L gesetzt. Im Bereich  $0.8V \leq U_e \leq 2.0V$  führten Einstreuungen zu teils starken Fluktuationen des Pegels am Aus- und Eingang.

**NAND als elementares Gatter** Die OR-Funktion konnte in positiver Logik erfolgreich implementiert werden. Interpretiert man die Implementierung mit drei NAND-Gattern in negativer Logik repräsentiert die Schaltung eine AND-Funktion auf Grund der Dualität zwischen AND und OR (deren Wahrheitstafeln gehen durch die Substitution  $1 \leftrightarrow 0$  ineinander über).

Die XOR-Funktion konnte ebenfalls in positiver Logik mit NAND-Gattern realisiert werden. Die XOR-Funktion summiert zwei 1-Bit Dualzahlen ohne Übertrag und Carry-Bit auf. Durch Erweiterung der Schaltung um eine AND-Funktion konnte das Carry-Bit ausgegeben werden. Die resultierende Schaltung wird als Halbaddierer bezeichnet und ist in der Lage die niedrigstwertigen Bits (*least significant bit*, **lsb**) zweier Dualzahlen zu addieren.

**Multiplexer** Der 2-zu-1-Multiplexer erlaubte es durch setzen des Bits an  $E_3 \equiv S$  das Signal von  $E_2$  and den Ausgang zu legen während im anderen Fall  $E_1$  auf den Ausgang gelegt wurde. Ein Multiplexer lässt sich also bildlich als umgekehrte "Datenweiche" verstehen und ist als solche elementarer Bestandteil digitaler Informationsverarbeitung.

**Dekodierschaltungen** Der mit sechs NAND-Gattern implementierte 2-Bit Decoder erlaubte eine Decodierung von 2-Bit Dualzahlen, deren **lsb** an  $E_1$  und **msb** an  $E_2$  anzulegen waren. Interpretierte man den Pegel L als aktives Signal entsprachen die Bezeichnungen der Ausgänge den dekodierten Zahlen in Dezimaldarstellung.

In Verbindung mit einem Festwertspeicher lässt sich ein 4-Bit Decoder zu einem Treiber für 7-Segment-Anzeigen verschalten. Dieser erlaubt die Ansteuerung einer solchen Anzeige, indem die gewünschte Zahl zwischen 0 und 9 als Dualzahl an den 4-Bit Eingang angelegt wird.

## A. Messwerte

.

.