

# Elektronikpraktikum – Versuch 7: Logische Schaltungen

Jonas Wortmann<sup>\*1</sup> and Angelo V. Brade<sup>†1</sup>

<sup>1</sup>Rheinische Friedrich-Wilhelms-Universität Bonn

17. September 2024

---

<sup>\*</sup>s02jwort@uni-bonn.de

<sup>†</sup>s72abrad@uni-bonn.de

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Theorie</b>	<b>1</b>
2.1	Boolische Algebra und Schaltfunktionen . . . . .	1
2.2	Flip-Flops . . . . .	1
2.3	Schrieberegister und Zähler . . . . .	1
2.4	Aufbau von elektronischen Logikschaltungen . . . . .	1
<b>3</b>	<b>Vorausgaben</b>	<b>2</b>
<b>4</b>	<b>Auswertung</b>	<b>4</b>
4.1	e) & f) . . . . .	4
4.2	g) . . . . .	4
<b>5</b>	<b>Fazit</b>	<b>5</b>

## 1 Einleitung

Dieser Versuch ist die Einführung in die boolische Elektrotechnik, wobei wir boolische Algebra nutzen, um logische Operationen zu konstruieren und z.B. Flip-Flops bauen, mit denen wir ein Schreiberegister und Zähler bauen. Dabei basiert alles darauf, dass hohe Spannungen einer logischen 1 zugeordnet werden und niedrige Spannungen einer logischen 0 zugeordnet werden. Von hier aus wird die grundlegende Informatik praktisch aufgebaut.

## 2 Theorie

### 2.1 Boolische Algebra und Schaltfunktionen

Bei digitalen Schaltelementen gibt es im allgemeinen mehrere Eingänge und einen Ausgang, wobei alle Signal als 0 oder 1 interpretiert werden. Das Verhalten lässt sich mit Schaltfunktionen beschreiben, die von der Menge der Eingangsvariablen auf die Menge der Ausgangsvariable abbildet. Um diese Funktionen darzustellen werden Funktions- tafeln verwendet, die alle Kombinationen an Eingängen, sowie die zugeordneten Ausgang angeben.

Für eine Eingangsvariable sind die folgenden operationen möglich:

<b>Identität</b>	$p(x) = x$
<b>Komplement</b> oder <b>Negation</b>	$p(x) = \bar{x}$
sowie konstant 1	$p(x) = 0$
und konstant 0	$p(x) = 1$

Für Funktionen mit zwei Eingangsvariablen, sog. elementare Funktionen, sind diese möglich:

$x_1$	$x_2$	<b>Konjunktion UND</b> $x_1 \cdot x_2$	<b>Disjunktion ODER</b> $x_1 + x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Tabelle 1: Elementare Funktionen

Nun lässt sich jede boolesche Funktion  $F$  als Summe von Mintermen, die sog. disjunktive Normalform, oder als Produkt von Maxtermen, die sog. konjunktive Normalform, ausdrücken. Hierbei ist eine Minterm  $m$  eine boolesches Produkt aus jeder Eingangsvariable oder ihrem Komplement, wobei diese jeweils nur einmal auftreten. Genauso ist ein Maxterm  $M$  eine boolesche Summer aus jeder

Eingangsvariable oder ihrem Komplement, wobei diese jeweils nur einmal auftreten. Hier sind gebräuchliche Schaltfunktionen dargestellt:

Symbol	BOOLEsche Funktion	Bezeichnung	
		deutsch	englisch
	$a \cdot b$	UND	AND
	$a + b$	ODER	OR
	$\overline{a \cdot b}$	NICHT - UND	NAND
	$\overline{a + b}$	NICHT - ODER	NOR
	$\bar{a}$	INVERTER	NOT

Abbildung 1: Gebräuchliche Schaltfunktionen

### 2.2 Flip-Flops

Die einfachste Möglichkeit ein Signal zu speichern, ist mithilfe eines Flip-Flops.

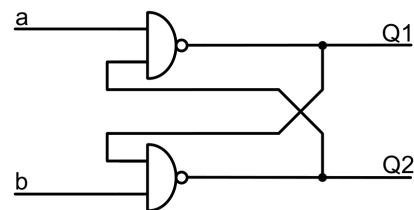


Abbildung 2: Flip-Flop

Ist  $a = b = 1$ , so ist das signal gespeichert und es lässt sich durch belegen mit einer 0 löschen. Hierbei ist dann  $Q_2 = \bar{Q}_1$  gespeichert.

### 2.3 Schreiberegister und Zähler

Nun kann man mithilfe von FFs ein Schreiberegister bauen, dass mit jedem Takt den Inhalt des  $i$ -ten FFs in das  $(i + 1)$ -te FF schreibt. Neben dem gibt es dann noch den Dualzähler. Er erzeugt in ansteigender Reihenfolge mit jedem Takt Dualzahlen.

### 2.4 Aufbau von elektronischen Logikschaltungen

Um nun Logikschaltungen zu konstruieren, wird eine Minimalspannung, ab der das Signal als 1 interpretiert wird, und eine Maximalspannung, unter der das Signal als 0 interpretiert wird, definiert. Diese Schwellspannungen lassen sich in Übertragungskennlinien darstellen, hier in Abb. 3.

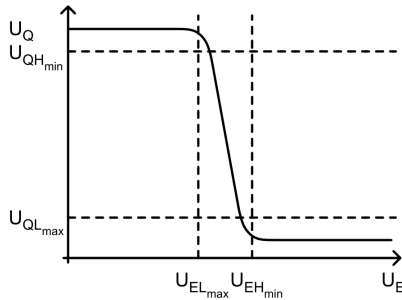


Abbildung 3: Übertragungskennlinie eines Inverters

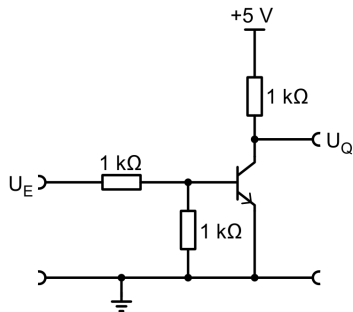


Abbildung 4: Schaltung eines Inverters

Z.B. ein Inverter kann nun mit Transistoren realisiert werden. Hier in Abb. 4 dargestellt. Diese Schaltung kann mit z.B. Dioden erweitert werden, um mehr Eingänge zu erhalten. Dies wird allerdings heutzutage nicht mehr mit Dioden, sondern direkt mit Transistoren gemacht, da diese deutlich kleiner hergestellt werden können. So kann auch ein CMOS-Gatter realisiert werden, welches allein aus Transistoren besteht und somit extrem klein gebaut werden kann. Dies ist hier in Abb. 5 dargestellt.

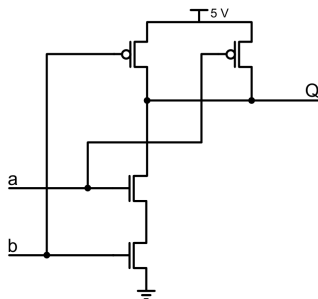


Abbildung 5: Schaltung eines CMOS-Gatter

### 3 Voraufgaben

#### A

Wenn man  $n$  Eingangsvariablen hat, gibt es ohne Redundanzen  $2^{2n}$  Schaltfunktionen.

#### B

Es werden folgende Ausdrücke mit einer Funktionstafel überprüft.

$$\begin{array}{lll} a + 1 = 1 & a \cdot 1 = a & a + \bar{a} = 1 \\ a + 0 = a & a \cdot 0 = 0 & a \cdot \bar{a} = 0 \end{array}$$

Man erkennt in Tab. 2, dass alle Ausdrücke korrekt sind.

Funktion $f$	$a$	$\bar{a}$	$f(a, \bar{a})$
$a + 1$	0	1	1
	1	0	1
$a + 0$	0	1	0
	1	0	1
$a \cdot 1$	0	1	0
	1	0	1
$a \cdot 0$	0	1	0
	1	0	0
$a + \bar{a}$	0	1	1
	1	0	1
$a \cdot \bar{a}$	0	1	0
	1	0	0

Tabelle 2: Funktionstafel

#### C

Nun wird das Distributivgesetz und das Gesetz von De-Morgan mithilfe von Funktionstafeln bestätigt. Diese sind in Tab. 3 und 4 dargestellt.

$a$	$b$	$c$	$a + b$	$(a + b) \cdot c$	$a \cdot c$	$b \cdot c$	$a \cdot c + b \cdot c$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	0	1	1
1	0	0	1	0	0	0	0
1	0	1	1	1	1	0	1
1	1	0	1	0	0	0	0
1	1	1	1	1	1	1	1

Tabelle 3: Funktionstafel für das Distributivgesetz

$a$	$b$	$\bar{a}$	$\bar{b}$	$a \cdot b$	$\overline{a \cdot b}$	$\bar{a} + \bar{b}$	$a + b$	$\overline{a + b}$	$\bar{a} \cdot \bar{b}$
0	0	1	1	0	1	1	0	1	1
0	1	1	0	0	1	0	1	0	1
1	0	0	1	0	1	1	1	0	0
1	1	0	0	1	0	0	1	0	0

Tabelle 4: Funktionstafel für das DeMorgan-Gesetz

**D**

$$f(a, b) = (a + b) \cdot \overline{(a \cdot b)} = \quad (1)$$

Der Ausdruck folgt der Tabelle, allerdings beschreibt  $\overline{(a \cdot b)}$  nicht die Tabelle.

**E**

Da wir für jede Variable entweder die normale Form oder die negierte Form und somit zwei Zustände haben, und wir jede Kombination durchgehen wollen, kann man binär hochzählen, wobei dann 0 für normal und 1 für negiert steht.

Binäre Zahl	entsprechender Minterm
000	$a \cdot b \cdot c$
001	$a \cdot b \cdot \bar{c}$
010	$a \cdot \bar{b} \cdot c$
011	$a \cdot \bar{b} \cdot \bar{c}$
100	$\bar{a} \cdot b \cdot c$
101	$\bar{a} \cdot b \cdot \bar{c}$
110	$\bar{a} \cdot \bar{b} \cdot c$
111	$\bar{a} \cdot \bar{b} \cdot \bar{c}$

Tabelle 5: Mintermbestimmung mithilfe von binärem Zählen.

**F**

Sobald  $a$  und  $b$ , 1 sind, sind die Ausgänge nicht mehr eindeutig. Für sonstige Zustände sind die Ausgänge eindeutig.

$a$	$b$	$Q_1$	$Q_2$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0 oder 1	0 oder 1

Tabelle 6: Funktionstafel von einem Flip Flop

**G**

Um einen 4-Bit-Schreibregister zu konstruieren, müssen vier D-FF hintereinander geschaltet werden, welche synchron getaktet werden. Dieser ist in Abb. 6 dargestellt.

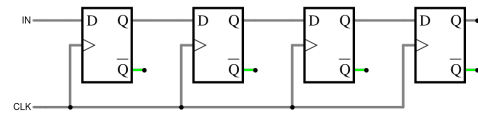


Abbildung 6: 4-Bit Schreibregister

**H**

Ein paralleles Schieberegister wird nach Abb. 7 konstruiert.

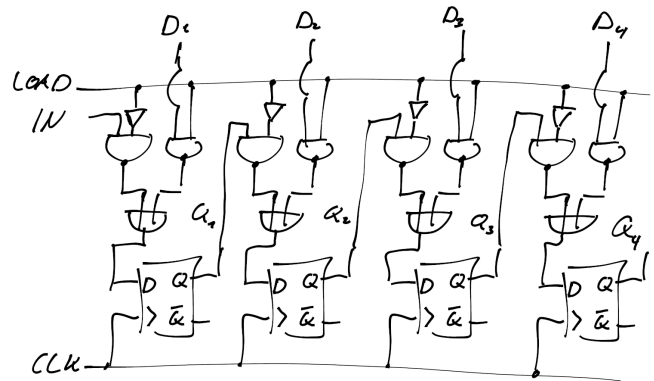


Abbildung 7: parallel 4-Bit Schreibregister

**I**

Ein Dualzähler wird nach Abb. 8 konstruiert.

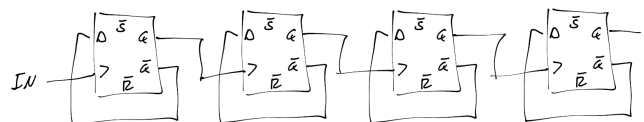


Abbildung 8: parallel 4-Bit Dualzähler

**J**

Die Funktion der Schaltung entspricht einer NOR-Schaltung. Die beiden Dioden dienen als Eingangssignal, das aktiviert ist, sobald einer der beiden Eingänge aktiv ist. Somit erhalten wir eine OR-Schaltung. Nun ist hinter den Dioden ein Invertierer geschaltet, welcher das Signal invertiert und so die Schaltung zu einer NOR-Schaltung vervollständigt.

Da wir eine Basis-Spannung von ca. 1.8 V haben und über Basis-Emitter  $U_{BE} = 0.7 \text{ V}$  abfällt können wir mit

$$I_B = \frac{1.8 \text{ V} - 0.7 \text{ V}}{1 \text{ k}\Omega} - \frac{0.7 \text{ V}}{1 \text{ k}\Omega} \quad (2)$$

$$= 0.4 \text{ mA} \quad (3)$$

zeigen, dass der Ausgangspegel korrekt ist.

## K

Es fließt Strom, aber es gibt probleme.

Muss noch gemacht werden!!!!!!1

## L

Beim CMOS-Inverter fließt bei geringer anliegender Spannung Strom, sodass beim Ausgang Spannung anliegt. Bei hoher Spannung fließt kein Strom, sodass keine Ausgangsspannung anliegt.

## M

Die Zeichnung realisiert ein NAND-Gatter. Es ist immer logisch 1, außer beide Eingänge sind logisch 1.

## N

Ein Übertrag eines Halbaddierers ist ein XOR und die Summe eine AND.

## O

Die Funktionstafel eines Volladdierers mit vorherigem Bit ist in Abb. 7 zu sehen.

$a$	$b$	$S$	$\bar{U}$	$\bar{U}_{i-1}$
0	0	0	0	0
0	0	1	0	1
0	1	1	0	0
0	1	0	1	1
1	0	1	0	0
1	0	0	1	1
1	1	0	1	0
1	1	1	1	1

Tabelle 7: Funktionstafel eines Volladdierers mit vorherigem Bit

## P

In Abb. 9 ist ein Volladdierer dargestellt.

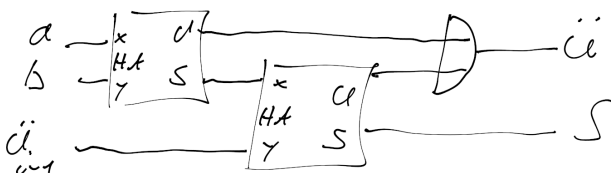


Abbildung 9: Volladdierer aus zwei Halbaddierern.

## Q

In Abb. 10 ist ein serielles Addierwerk dargestellt.

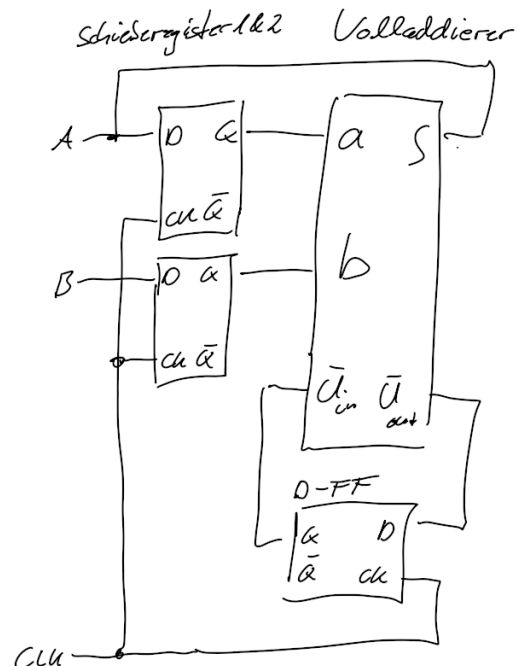


Abbildung 10: Serielles Addierwerk

## 4 Auswertung

### 4.1 e) & f)

Mit dem seriell oder parallel ladbaren n-Bit Schieberegister lässt sich eine n-Bit Zahl speichern und wieder ausgeben. Die Schaltskizzen sind in den Voraufgaben zu finden. Im Versuch wurde erfolgreich ein serielles 4-Bit und parallels 3-Bit Schieberegister aufgebaut. Das 3-Bit Schieberegister wurde auch als Ringschieberegister aufgebaut, indem der letzte Ausgang wieder an den Eingang angeschlossen wird, was es möglich macht, die Zahl „im Kreis“ durch das Register zu schieben. Verschiedene Zahlen konnten gespeichert und ausgelesen werden; die Schaltung (oder Speichern und Auslesen der Bits) funktioniert nach dem FIFO Prinzip.

### 4.2 g)

Mit den beiden Halbaddierern kann ein Volladdierer gebaut werden, welcher, durch Verwendung zweier serieller 3-Bit Ringschieberegistern zu einem Addierwerk verbaut werden kann. Dieses wurde bereits in den Voraufgaben besprochen. Dieses Addierwerk kann zwei 3-Bit Zahlen addieren, allerdings nur bis maximal 111 darstellen. Das Limit ist also die Größe des Ringschiebers.

Eine Zahl kann durch umlegen eines Schalters in einen Ringschieber eingelesen werden. Indem man die Ringschieber taktet „rutscht“ der Bit von der dritten Stelle ( $3_2 = 4_{10}$ ) zur zweiten ( $2_2 = 2_{10}$ ) und zur ersten Stelle ( $1_2 = 1_{10}$ ). Durch weitere drei Takte können diese beiden Zahlen addiert werden. Das Addieren funktioniert so, dass immer jeweils ein Bit aus den Ringschiebern mit dem Volladdierer addiert werden und der Übertrag wieder in den Eingang des Volladdierers eingeführt wird. Das Ergebnis wird im ersten Ringschieber angezeigt (die eingegebenen Zahlen gehen dabei verloren). Einige Beispiele sind

$$001 + 000 = 001 \quad 011 + 001 = 100 \quad 100 + 011 = 111. \quad (4)$$

## 5 Fazit

In diesem Versuch wurde die Übertragungskennlinie eines Invertierers (aufgebaut aus CMOS) gemessen. Mit Hilfe

dieser CMOS wurde ein NAND Gatter gebaut, für welches wieder die Übertragungskennlinie gemessen wurde. Beide Kennlinien haben den theoretischen Erwartungen entsprochen.

Danach wurde Schritt für Schritt ein serielles 3-Bit Addierwerk aufgebaut. Angefangen mit einem XOR-Gatter aus NAND-Gattern zu einem Halbaddierer und dann einem Volladdierer, indem ein zweiter Halbaddierer gebaut wurde und die Überträge entsprechend zurückgeführt wurden. Mit einem seriellen 3-Bit Ringschieber konnten dann zwei verschiedene Zahlen eingelesen werden und durch drei mal takten (da es sich hier um 3-Bit Zahlen handelt) des Volladdierers addiert werden. Das Ergebnis wurde in einem Ringschieber gespeichert und konnte durch Abgreifen der Ausgänge angezeigt werden.