# Versuch 7: Logische Schaltungen

Leonie Dessau & Carla Vermöhlen

7. September 2025

## Inhaltsverzeichnis


**1 Einleitung** **2**

**2 Theorie** **2**

**3 Voraufgaben** **2**

3.1 A . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
3.2 B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
3.3 C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
3.4 D . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
3.5 E . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
3.6 F . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
3.7 G . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
3.8 H . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
3.9 I . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

**4 Fazit** **7**

**5 Anhang** **7**


## Vorbemerkungen

Dieses Protokoll wurde gemeinsam von Carla Vermöhlen und Leonie Dessau erstellt und (außer uns sind Fehler bei der Versionierung unterlaufen) zwei mal gleich abgegeben. Quellcode (auch LaTeX) verfügbar auf `https://github.com/byteOfWisdom/the-ep-cant-hurt-you`. Schaltbilder ohne explizite Quelle sind mit Tikz erzeugt, Diagramme ohne explizite Quelle mit Python (Oder gnuplot. Oder Julia.). Die Signaldiagramme wurden mit den csv Dateien aus dem Oszilloskop geplottet.

# 1 Einleitung

# 2 Theorie

# 3 Voraufgaben

## 3.1 A

*Wieviel verschiedene Schaltfunktionen von n Eingangsvariablen gibt es, wenn man nur Schaltfunktionen ohne Redundanzen betrachtet?*

Bei $n$ Eingangsvariablen, gibt es $2^n$ mögliche Eingangswerte. Da wir nach $B$ abbilden, gibt es pro Eingangswert je zwei mögliche Ausgangswerte. Da die Schaltfunktion die Abbildung aller möglichen Eingangswerte nach $B$ ist, ist jeder neue Satz an Ergebnissen eine eigene Schaltfunktion. Diese Liste an Ergebnissen kann als $2^n$ Stellige Binärzahl betrachtet werden. Somit gibt es $2^{2^n}$ Schaltfunktionen für $n$ Eingangsvariablen.

## 3.2 B

*Prüfen Sie die obigen Ausdrücke anhand einer Funktionstafel nach.*

| a | 1 | = |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 1 | 1 |

Tabelle 3.1: z.z.: $a + 1 = 1$

| a | 0 | = |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Tabelle 3.2: z.z.: $a + 0 = a$

| a | 1 | = |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |

Tabelle 3.3: z.z.: $a \cdot 1 = a$

| a | 0 | = |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |

Tabelle 3.4: z.z.: $a \cdot 0 = 0$

| a | = |
|---|---|
| 0 | 0 |
| 1 | 0 |

Tabelle 3.5: z.z.: $a \cdot \overline{a} = 0$

| a | = |
|---|---|
| 0 | 1 |
| 1 | 1 |

Tabelle 3.6: z.z.: $a + \overline{a} = 1$

## 3.3 C

*Prüfen Sie das Distributivgesetz und die Sätze von DeMORGAN mit einer Funktionstafel nach.*

| a | b | = |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Tabelle 3.7: Eine Wahrheitstabelle!

## 3.4 D

*Wie lautet der BOOLEsche Ausdruck für die EXKLUSIV-ODER-Funktion aus Beispiel 1? Formen Sie den Ausdruck um, bis nur noch die Schaltfunktion $\overline{a \cdot b}$ vorkommt*

XOR ist $(a \cdot \overline{b}) + (\overline{a} \cdot b)$
Zunächst einige Vorüberlegungen, wie sich UND und OR und NOT über NAND ausdrücken lassen:

$$x + y = \overline{(\overline{x} \cdot \overline{y})} \tag{3.1}$$

$$\overline{x} = \overline{x \cdot x} \tag{3.2}$$

$$x \cdot y = \overline{\overline{x \cdot y}} = \overline{\overline{(x \cdot y)} \cdot \overline{(x \cdot y)}} \tag{3.3}$$

Mit diesen darstellungen ist nun nur noch stumpfes Einsetzen nötig.

$$(a \cdot \overline{b}) + (\overline{a} \cdot b) = \overline{\overline{(a \cdot \overline{b})} \cdot \overline{(\overline{a} \cdot b)}} \tag{3.4}$$

Akzeptiert man, dass NOT sich durch NAND als Schaltkreis trivial bauen[1] lässt und somit in der NAND-Form nichts dagegen spricht NOT zu nutzen, so ist man num fertig. Besteht man darauf, wirklich *nur* NAND Operatrionen zu nutzen, muss man noch ein wenig mehr einsetzen (auch wenn es schwachsinnig ist):

$$\overline{\overline{(a \cdot \overline{b})} \cdot \overline{(\overline{a} \cdot b)}} = \overline{\overline{(a \cdot \overline{(b \cdot b)})} \cdot \overline{(\overline{a \cdot a} \cdot b)}} \tag{3.5}$$

## 3.5 E

*Schreiben Sie alle Minterme von 3 Eingangsvariablen auf. Vergleichen Sie die Anzahl der verschiedenen Minterme mit der Zeilenzahl einer Funktionstafel für 3 Eingangsvariable. Wie wird man die Minterme sinnvollerweise nummerieren?*

Die Miniterme sind:

$$\begin{gathered} \overline{a} \cdot \overline{b} \cdot \overline{c} \\ \overline{a} \cdot \overline{b} \cdot c \\ \overline{a} \cdot b \cdot \overline{c} \\ \overline{a} \cdot b \cdot c \\ a \cdot \overline{b} \cdot \overline{c} \\ a \cdot \overline{b} \cdot c \\ a \cdot b \cdot \overline{c} \\ a \cdot b \cdot c \end{gathered}$$

Was auch die Sinnvolle Numerierung ist, da dies dann analog zu der kanonische Darstellung von Binärzahlen ist (wobei 0 einer negierten Variable und 1 einer nicht negierten entspricht).

## 3.6 F

*Stellen Sie eine Funktionstafel (Eingänge a, b, Ausgänge Q1, Q2) dieses Flip-Flops auf. Starten Sie dazu mit beliebigen Zuständen für Q1 und Q2 und verfolgen Sie, wie sich die Ausgänge durch die Rückkopplung ändern. Für welchen Eingangszustand a, b gibt es mehrere Möglichkeiten für die Ausgänge?*

Sei initial Q1 auf $A1$ und Q2 auf $\overline{A1}$ gesetzt:

| b | a | Q1 | Q2 |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | $A1$ | $\overline{A1}$ |

Tabelle 3.8: Fall 1

[1]Signal and beide Eingänge eines NAND Gatters

Im Falle von $a \neq b$ kann der Zustand von Q1 und Q2 eindeutig bestimmt werden (siehe Tabelle) und hängt nicht vom Initialzustand ab. Ist $a = b = 1$, dann findet keine Rückkopplung statt, da der resultierende Zustand immer stabil ist und genau dem Anfangszustand entspricht.
Im Fall $a = b = 0$ werden Q1 und Q2 beide 1, dieser Zustand ist aber nicht stabil bei $a = b = 1$, in diesem Falle ist der Zustand nicht eindeutig definiert und würde in der Realität enweder oszillieren oder einen zufälligen Zustand annehmen.
Es gibt also für $a \neq b$ jeweils nur einen Zustand, während $a = b$ jeweils zwei mögliche Zustände hat.

## 3.7 G

*Zeichnen Sie ein 4-Bit-Schieberegister auf, das seriell geladen wird.*

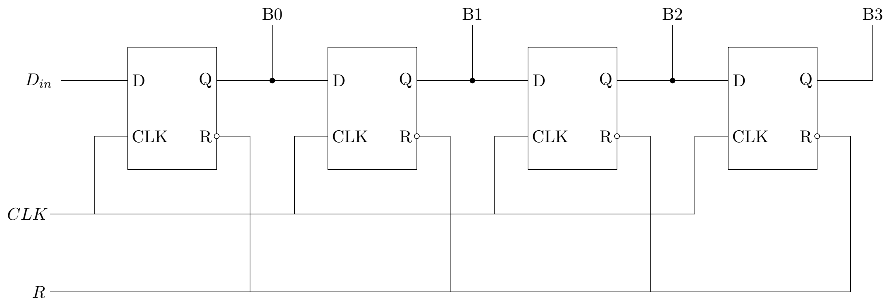


Abbildung 3.1: Serielles Shift-Register mit 4 Bit

## 3.8 H

*Entwerfen Sie ein 4-Bit-Schieberegister, das parallel geladen werden kann (d. h. alle Bits gleichzeitig, wenn eine Steuerleitung "LOAD" auf 1 ist). Benutzen Sie dazu die unten abgebildeten kombinierten Schaltelemente, die auch auf dem Schaltbrett zur Verfügung stehen.*

Es soll bei jedem Bit, der Eingang D immer entweder Q des vorherigen Bits oder der parallele Eingang $D_i$ sein. Formal ist dies $(Q_{n-1} \cdot \overline{\text{LOAD}}) + (D_i \cdot \text{LOAD})$. Es wird als ein negiertes LOAD Signal benötigt, welches hier einfach mittels einen NAND Gatters bereitgestellt wird.
Ein Schaltplan kann dann wie folgt aussehen:

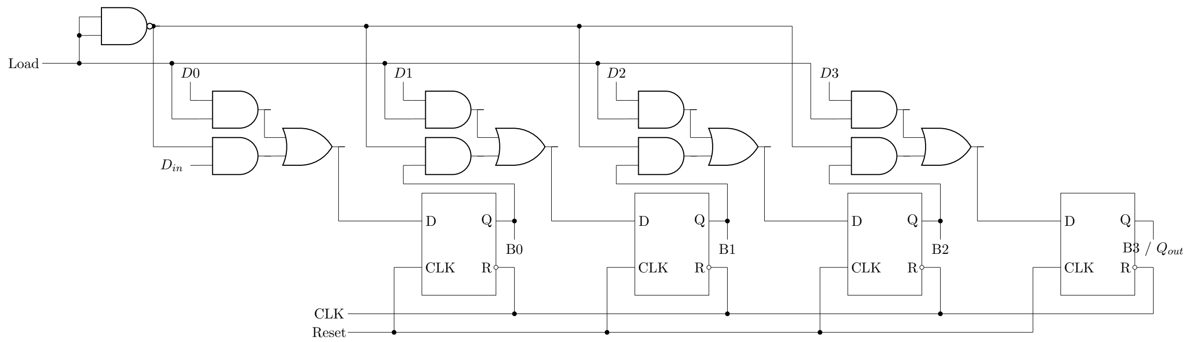


Abbildung 3.2: 4 Bit Shift Register mit der Option zur parallelen Befüllung über D0 bis D3, wenn Load 1 ist, ansonsten Seriell über $D_{in}$.

Das Auslesen kann dann entweder parallel über B1 bis B4 oder Seriell über $Q_{out}$ erfolgen.

## 3.9 I

*Entwerfen Sie einen 4-Bit-Dualzähler, bei dem der Ausgang eines FFs jeweils den Takteingang des nächsten FF steuert. Tip: Verbinden Sie bei jedem Flipflop Q mit D.*

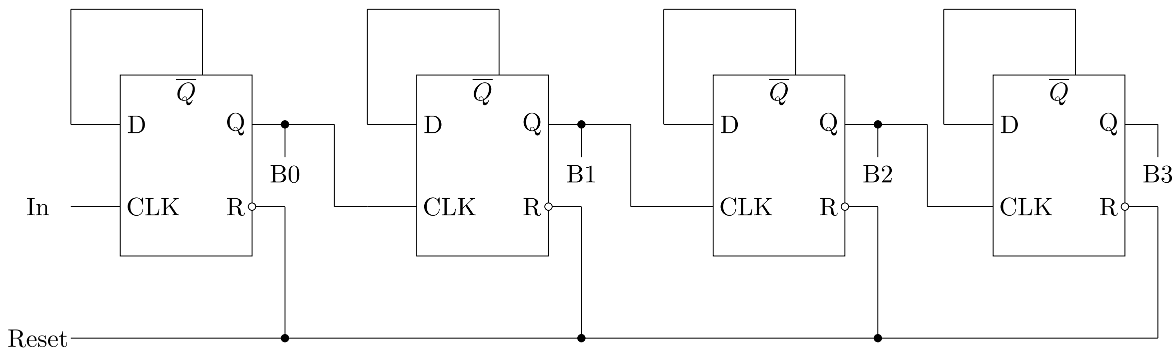


Abbildung 3.3: 4 Bit Dualzähler

# 4 Fazit

# 5 Anhang